A Knowledge-Based Engineering System Framework for the Development of Electric Machines

PhD candidate CHRISTIAN RIVERA
Director: JAVIER POZA LOBO
Co-director: GAIZKA UGALDE ROSILLO

Mondragon Goi Eskola Politeknikoa
Mondragon Unibertsitatea

Arrasate-Mondragón (Gipuzkoa), October 2019

Abstract

The new concept industry 4.0 is a great opportunity to improve the competitiveness in a global market for small-medium size electric machinery companies. The demand for electric motors have increased in the last decade especially due to applications that try to make a full transition from fuel to electricity. These applications encounter the need for tailor-made motors that must meet demanding requirements. Therefore, it is mandatory small-medium companies adopt new technologies offering customized products fulfilling the customers' requirements according to their investment capacity. Furthermore, simplify their development process as well as to reduce computational time to achieve a feasible design in shorter periods. In addition, find ways to retain know-how that is typically kept within each designer either to retrieve it or transfer it to new designers.

To support the aforementioned issue, a knowledge-based engineering (KBE) system framework for the development of electric machines is devised. The framework is encapsulated in the so-called KBV2-model comprising the standardized macro-level framework for electrical machine and the knowledge base generation process. This thesis describes this model and the integration of KBE applications with current industrial technologies such as Model-Based Systems Engineering (MBSE), Product Lifecycle Management (PLM), multiphysics and analytical design tools. This architecture provides capability to manage and automate tasks in the development process of electric machines.

The author of this work has opted to develop KBE applications following the minimum viable product principle. The KBE system framework herein presented is formalized through the experience and analysis of the development and implementation of the KBE applications. From which a guideline is provided following a sequential process in order to achieve a viable KBE system. To substantiate the process a KBE system is created that supports the development of electric motors for the elevator system industry.

Acknowledgements

First, I am grateful to the one and only true God in whom I depend for everything, for his grace and provision, making possible my dedication to this thesis. Undeserved grace which I received through repentance and trust in Jesus Christ alone. My heart, soul and strength are yours.

Thanks to my beautiful and virtuous wife, the helper suitable for me, for her support and dedication to us, especially to Lukas, my son. I never worried knowing that my son was always covered by his mom’s care. I am grateful with my wonderful parents for their support and love in my entire life. I want to say thank you to the rest of my family for their support, your help is always there even when I do not ask. Thanks to my family of faith, this would have not been possible without your love and support.

Thanks to my director PhD Poza and co-director PhD Ugalde for their guidance, patience and kindness. Their technical tutoring and generosity made possible the outcomes of this thesis, very clever and professional men that without their experience and knowledge I would have simply failed to finish this. Thanks to the rest of members of the electrical and computer engineering department for their help in several tasks. I am also grateful with members of the software engineering department like Daniel, Alain, Javi, and Aitor for answering my questions.

Many thanks, God bless you all.

TABLE OF CONTENTS

GLOSSARY

Acronyms

ALM	Application Lifecycle Management
BOM	Bill Of Materials
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CBSE	Component Base Software Engineering
CFD	Computational Fluid Dynamics
COTS	Commercial Off-The-Shelf Software
COTS	Commercial Off The Shelf Software
CPSR	Constant Power Speed Ratio
EMC	Electromagnetic Compatibility
FEM	Finite Element Method
FMI	Functional Mock-Up Interface
FW	Flux Weakening
KBV2	Knowledge Base generation for the V2-Model
GUI	Graphical User Interface
GUI	Graphical User Interface
ICARE	Illustration, Constraints, Activities, Rules, And Entities
IEC	International Electrotechnical Commission
INCOSE	International Council In Systems Engineering
IoT	Internet of Things
IPM	Interior Permanent Magnet Motor
KBE	Knowledge Based Engineering
KBES	KBE System
KBS	Knowledge Based Systems
KNOMAD	Knowledge Optimized Manufacture And Design
KPI	Key Performance Indicators
MBSE	Model Based Systems Engineering
MCDM	Multicriteria Decision Making

MDE	Model Design Engineering
MDO	Model Design Optimization
MEPS	Minimum Energy Performance Standards
MML	Moka Modelling Language
MOKA	Methodology and software tools Oriented to Knowledge based engineering Applications
MPSR	Maximum Power to Speed Ratio
MTPA	Maximum Torque Per Amperes
MTPV	Minimum Torque Per Voltage
MVP	Minimum Viable Product
NEMA	National Electrical Manufacturing Sssociations
OEM	Original equipment manufacturer
OMG	Object Management Group
OPM	Object Process Methodology
PDM	Product Data Management
PLM	Product Lifecycle Management
PWM	Pulse Width Modulation
RAMS	Reliability, Availability,Maintainability, and Safety
RTD	Resistance Temperature Detectors
SE	Systems Engineering
SME	Small Medium Enterprise
SysML	Systems Modeling Language
TOPSIS	Technique for order preference by similarity to ideal solution
TRL	Technology Readiness Levels
UI	User Interface
UML	Unified Modeling Language
VDI	Verein Deutscher Ingenieure
WASPAS	Weighted Aggregated Sum Product Assessment
WPM	Weighted Product Model
WSM	Weighted Sum Model
XMI	XML Metadata Interchange
XML	Extensible Markup Language

Variables

Symbol	Units	Definition
X	[m]	Distance
Qn	[kg]	Load of cabin
Tara	[kg]	Mass of cabin
Cont	-	Counterweight coefficient
$\mathit{Ncables}$	-	Number of cables
$\mathit{dens\_c}$	[kg/m^3]	Density of cables
susp	-	Roping ratio
Dp	[m]	Pulley diameter
vel	[m/s]	Velocity
acel	[m/s^2]	Acceleration
Jerk	[m/s^3]	Jerk
Conex	-	Starts per hour
fm	-	Duty cycle
$\mathit{Rend\_h}$	-	Hole efficiency
$\mathit{Rend\_m}$	-	Machine efficiency
Jmf	[kg·m^2]	Machine inertia
Jp	[kg·m^2]	Pulley inertia
$\mathit{factor\_Pmultiple}$	-	Weight coefficient
Trated	[Nm]	Rated torque
Tmax	[Nm]	Maximum torque
Trms	[Nm]	RMS torque
Nrpm	[RPM]	Speed
Irated	[A]	Rated current
Vlimit	[V]	Voltage limit
eff	-	Efficiency
pf	-	Power factor
$\mathit{dTripple}$	[pu]	Torque ripple per unit amplitude (for a given harmonic)
Lstk	[mm]	Length of stack
Dse	[mm]	Stator external diameter
dTcu	[°C]	Copper temperature rise
dTm	[°C]	Magnet temperature rise
Cost	[€]	Cost in euros
EMF	[V]	Electromagnetic force
Lin	[mm]	Length of stack input for cost function
Zin	-	Number of turns input for cost function
$\mathit{CuWeight}$	[kg]	Total weight of copper

SteelWeight	[kg]	Total weight of steel
MagnetWeight	[kg]	Total weight of magnets
LinMin	[mm]	Minimum length of stack
LinStep	[mm]	Incremental step of length of stack
LinMax	[mm]	Maximum length of stack
abscissa	-	Abscissa
kk	-	Number of layers
m	-	Number of phases
Ks1	-	Skew factor fundamental
Kl	-	Stacking factor
Ar	[mm^2]	Slot area
Fr	-	Filling factor
ns	-	Coil span
rho	[Ωm]	Resistivity
Nref	-	Number of turns per coil reference
Lfcref	[mH]	Leakage inductance reference
Co_Br	-	Magnet temperature coefficient
coef_t	-	Copper temperature coefficient
Delta_T_cobre	[°C]	Initial value of copper temperature rise
Delta_T_iman	[°C]	Initial value of magnet temperature rise
dTcu_iter_tolerance	[%]	Tolerance of acceptance for copper temperature rise
Kov	-	Overlapping factor
dens_cu	[kg/dm^3]	Density of copper
dens_steel	[kg/dm^3]	Density of steel
dens_magnet	[kg/dm^3]	Density of magnet
lambdaWASPAS	-	Value [0,1] for index calculation
In	[A]	Rated current
Imax	[A]	Current for maximum torque
Imax90	[A]	Current for maximum torque at 90% rated speed
Tn	[Nm]	Rated torque
Tmax	[%]	Maximum torque percentage with respect to rated torque
Tmax@90	[Nm]	Maximum torque at 90% rated speed
Tmax@90	[%]	Maximum torque percentage with respect to rated torque at 90% rated speed
VLL_n	[V]	Line to line voltage
VLL_max	[V]	Maximum Line to line voltage for Tmax
VLL_max@90	[V]	Maximum Line to line voltage for Tmax at 90% rated speed
Rs	[Ω]	Stator resistance
Lq	[mH]	Inductance
Lfc	[mH]	Leakage inductance

$Trms$	[%]	RMS torque percentage with respect to rated torque
$Irms$	[A]	Current for RMS torque
$Tmec$	[Nm]	RMS shaft torque
$Pmec$	[W]	Mechanical power
Ps	[W]	Electrical power
Pj	[W]	Joule losses
Ph	[W]	Core losses
Pb	[W]	Bearing losses
ρ	[W/dm^3]	Power density
$\Delta Tcu@Trms$	[°C]	Copper temperature rise for RMS torque
$\Delta Tm@Trms$	[°C]	Magnet temperature rise for RMS torque
η	[%]	Efficiency
Acu	[mm]	Cross section of magnet wire
Drm	[mm]	Mean slot diameter
Dsr	[mm]	Slot diameter (outer)
$Lesp$	[mm]	Length of turn
$Lend$	[mm]	End winding length
$WireDiam1$	[mm]	Wire diameter 1
$WireParallel1$	[mm]	Number of wires in parallel for WireDiam1
$WireDiam2$	[mm]	Wire diameter 2
$WireParalell2$	[mm]	Number of wires in parallel for WireDiam2
Nph	-	Number of turns per phase
N	-	Number of turns
Nb	-	Number of coils
Qs	-	Number of slots
wd	[mm]	Tooth width
Rso	[Ω]	Resistance at ambient temperature
Vcu	[m^3]	Volume of copper
Mcu	[kg]	Weight of copper
$Br0$	[T]	Magnetic field at ambient temperature

Chapter 1

INTRODUCTION

The scope of this thesis is the creation of a knowledge based engineering system framework for the development of electric machines. The purpose of this framework is to allow developing KBE systems for the electric machine industry providing capabilities such as development process traceability, knowledge management, automation of repetitive tasks and intelligent support.

This work presents the framework providing substantiation of the process for each stablished step. Background and review of current work is provided in each chapter. Depending the devised KBE project the framework provides two ways of developing KBE applications through the micro-process and macro-process knowledge representation. The KBE system framework is encapsulated in the so-called KBV2-model. The model consists of a knowledge base generation part and the standardized processes for the development of electric machines. Use case examples with comparison of time consumption by using and not using the KBE system are presented.

This chapter provides a general background of the work, with focus on KBE and describing the issues that the devised KBE system framework wants to tackle as well as the objectives set in this work. Furthermore, the publications in journals and conferences are introduced.

1.1 Background

Electric machines (motors and generators) are key elements for efficient and sustainable use of energy. The vast majority of electrical power generation systems (except photovoltaic and fuel cells) use an electric generator. Likewise, two thirds of the electrical energy of industrial processes is processed by electric motors. This thesis will be framed in the development of new methodologies, processes and design tools in order to globally improve the competitiveness of the electric machinery sector. The approach of the thesis seeks to make a generic improvement to the design processes, so that in the future it can be particularized to the different specific applications of electric machine. Although electric machines falls within a mature technology, due to the latest increases in energy costs and the needs of energy sustainability, an increasing replacement of standardized machines with new solutions that incorporate technological innovations in design, materials, manufacturing, control and maintenance have been carried out. The main factors of the market that are allowing a technological change are the following.

1. Need to increase energy efficiency in systems powered by electric motors. Current standards, policies and regulations for electric motors have introduced higher level of efficiencies for motors in the market. For instance, EC No4/2014 since 2017 states that motors from 0.75W-375W should be IE3 or IE2 with variable speed drives. Eco-design of electric motors is now a usual practice in industry. New technologies have emerged due to this. For instance, change of aluminium to copper in rotor structures, extension of the use of reluctance and permanent magnet motors, and direct drive technologies especially in low speed applications.

2. New renewable resources for power generation. The progressive introduction of renewable resources for electricity generation is leading to the need of new designs to optimized electric generators for varying conditions and for important restrictions in terms of working environments, weights, sizes, and maintenance needs. Innovative designs in electric generators are facilitating the competitiveness of these new generation resources. For instance, in wind generation high efficient permanent magnet generators are being implemented.

3. Electromobility boost. Due to progressive incentives for hybrid and electric vehicles, impulse of rail transport, and new forms of urban mobility, electric machines have increased their implementation in this sector. In order to achieve the cost and performance targets demanded by the traction sector, technical developments are required in electric machine architecture, integration, materials and supporting areas. For instance, electric motors requires to be coupled with the transmission, as well as to be improved in insulation, windings and manufacturing processes. In addition, it requires advanced controls to manage vibro-acoustic issues and efficiency optimization.

1.2 Competitive positioning of the electric machinery sector against key market drivers

This section presents current technology and market drivers in the electric machinery industry.

1.2.1 Technology and key market drivers

From the general study of the different current and future applications, the following are the relevant technological drivers in the new trends of electric machines.

- Development of high efficiency motor:
    - With current induction motor technology or permanent magnets.
    - With new reluctant or hybrid reluctant-magnet technology.
- Integration of the electronic drive into the motor.
- Integration of the motor into the application.
    - Increase in the use of direct-drive drives.
    - Motor design optimized in size/shape to the application requirements.
    - Performance design optimized to the specific requirements of the application
- Technological solutions with high reliability and low life cycle cost.
- Electric drive with greater intelligence that allows the system to operate at the points of minimum consumption and that diagnoses or predicts failures of the drive and application.

All these factors should be undertaken while maintaining competitiveness in development and manufacturing costs. The European small / medium sized electric machinery industry faces two tough challenges.

• Great competition from emerging countries that can manufacture at lower labor costs.

• Great competition from large electrical equipment industries (Siemens, ABB, GE) which, due to their large size, have a great capacity for technological development and investments in manufacturing.

1.2.2 Challenges for electric machines advanced design frameworks.

In this thesis, a representative SWOT analysis from the point of view of a small/medium-sized enterprise (SME) in Europe is developed by grouping the technology and market factors, see. Fig. 1-1.

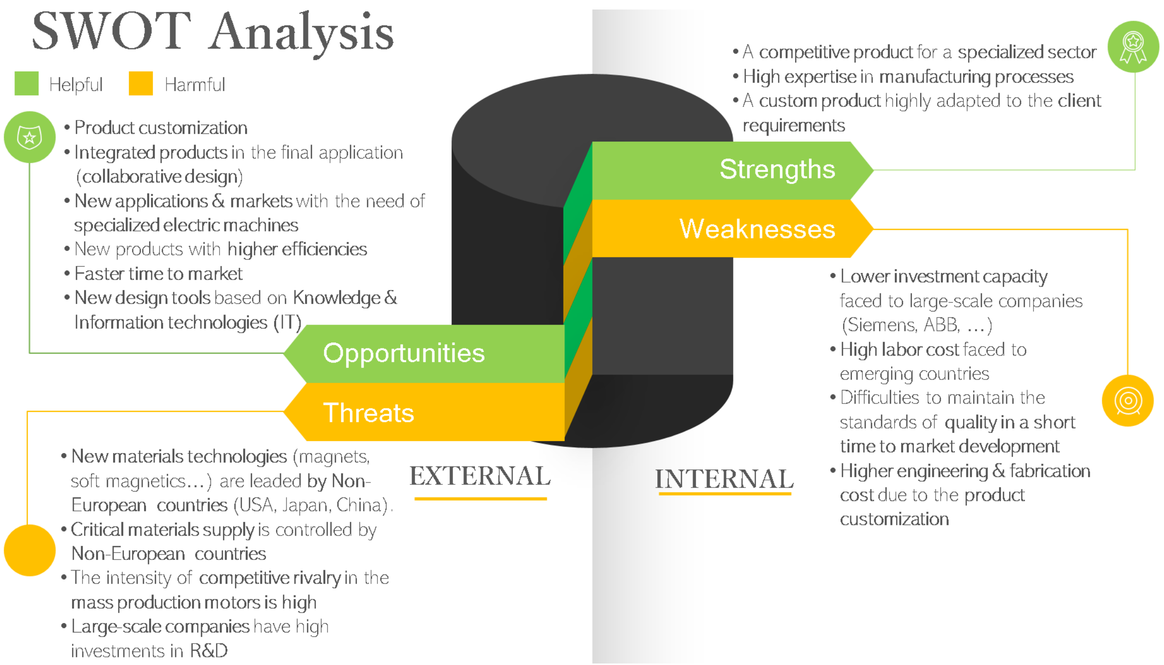


Fig. 1-1. A representative SWOT analysis for the European SME in the electric machine sector.

In order to improve the competitive positioning for a SME of electric machines, this thesis proposes the improvement of design procedures responding to market opportunities as well as the improvement of competitive weaknesses. Next, on each aspect of the carried out SWOT analysis, the opportunities for enhancing the development of a new electric machine design framework are detailed.

1.2.2.1 Strengths

Context:

The current medium and small European electric machine enterprises have been maintained in the market thanks to their specialization and to their offer of customized products to the client, with a highly optimized manufacturing process. Taking as an example, the electric machine industry of the Basque Country, the following companies represent the specialization of the sector: Large Machines (Indar / Alconza), Elevation (Lancor, Zardoya-Otis, CEG), Special Machinery (Obeki, Leberri).

Challenges for the new design framework:

The accumulated experience in design and manufacturing of companies is high; with diverse trajectories, in some cases they have had to undergo internal transformations of product and structure to survive in a business sector context of high competition. The new design framework has to take advantage of this experience and transform it into knowledge, becoming a future asset for the company. An important characteristic of traditional electric machine enterprises is that there is a lot of implicit knowledge distributed in specific people. The transmission of this knowledge through training to new employees entails non-negligible efforts in time and dedication, requiring a long-term strategy. There is also the risk of loss of knowledge in the case of critical people leaving the company. A challenge for the "new design frameworks" is allowing capturing and systematizing the implicit knowledge of the different people and processes of the organization. In this sense, design methodologies based on Knowledge Based Engineering (KBE) can be very useful.

1.2.2.2 Weaknesses

Context:

The small and medium size of the enterprises makes hard to cope with large investments to develop a fully automated manufacturing process. Similarly, a small and changing series production makes risky the large investments in the automation of the assembly line. Higher personnel costs compared to emerging countries aggravate the investments for a complete automation. Furthermore, the new context of product customization means that engineering costs and investments for new machinery and tools are high. This should be taken into account for the launch of a new product. Another important aspect is the quality assurance in a context in which new products have to be quickly introduced into the market. Hence, having shorter development and validation times.

Challenges for the new Design Framework:

• The development cost of the new design system should be appropriate to the size and needs of the company. Similarly, the new design system should significantly reduce engineering costs when facing product customization. In this area, the use of design techniques based on MBD (Model Based Design) models, in which the design is virtualized can reduced the costs of design modifications.

• The validation of a new product requires considerable time and dedication. In this regard, it is very useful to use the new **Modular Design** concepts in which each module (which may include internal modifications) is individually validated, and during customization only the correct integration to the system is validated.

• In the field of modular design, it is important to reduce investments in the manufacturing of the customized products so that each module be manufactured with a minimum modification of machinery and tools. In this regard, the New Framework should include, from the beginning of the design process, the consideration of the final manufacturing process to be used, being a framework that enables the new **Design for manufacturing** capabilities.

1.2.2.3 Threats

Context:

- The cost of materials represents about 40 ÷ 60% of the manufacturing cost of the electric machine. Within a highly competitive market, increases in the cost of raw materials can jeopardize the profitability margin of the machine manufacturer due to the difficulty to pass on this increase in cost directly to the final product of the client. In addition, manufacturers try to reduce material stocks to reduce financial costs. Certain critical materials could cause supply problems making impossible to respond in time to increases in customer orders. For instance, the critical supply of rare earth materials, such as Neodymium (Nd) and Disproxium (Dy), which are used in permanent magnets of higher magnetic energy density. The extraction of these materials is mainly located in China. Geopolitical positions put the risk of abrupt price increases and temporary lack of supply of these materials (as happened during the rare earth crisis during 2011).

- In response to this need for critical materials. Since 2010 the different governments, led by the USA, Japan, and China, are financing expensive research programs for the development of high-tech magnetic materials, in which due to advanced manufacturing, they can maximize the use of raw materials. The appearance on the market of these new magnetic materials can make the current products of European manufacturers technologically obsolete.

- It is needed to take into account the high R&D efforts of the large companies that manufacture electric machines, which have a great potential for new developments in new motors and manufacturing technologies threatening the current competitiveness of SME manufacturers.

Challenges for the new Design Framework:

- The design scheme should enable the development of new innovative value-added proposals. These proposals may come with the development of new concepts of electric machines better adapted to a given application, the use of lower cost manufacturing processes, a machine design that allows the use of new materials that are more competitive, etc.

- The development of an advanced design tool that reduces less productive engineering times, and maximizes the potential for innovation of designers would make possible to obtain greater profitability for the R&D efforts that the company can dedicate.

- It shows vital importance, the collaboration with agents that develop new materials; either in the collaboration of European joint development projects sponsored by the European Union (e.g. EIT Raw Materials funding Project) or through joint development with companies supplying raw materials (which in their initial stages of development require implementations into applications to demonstrate the potential of its new materials).

- The new design framework should make possible the incorporation of competitive surveillance of new materials and new design and manufacturing technologies. It should also provide a mechanism for collaborative work with suppliers and agents for the development of new materials.

1.2.2.4 Opportunities

Context:

It should be noted that the direct clients that initially purchase electric machines are industries/generation plants (usually large companies) that use the electric machines to drive/take advantage of the energy in their facilities. Alternatively, equipment manufacturing companies (industrial machinery, household appliances, means of transportation, etc.) that integrate the electric machine in their final applications. The end user of the equipment pays attention mainly to the overall performance, and the requirements of the electric machine remain in the background as long as there are no operational problems (breakdowns, abnormal noises, high consumptions, etc.) that are attributable to the electric machine. Therefore, within the demand for a specialized electric machine product, the client profile is set either by a large industrial/generation company or by a manufacturer of reference equipment in its sector. For these clients, product customization and/or the improvement of the integration of the electric machine in their equipment can be a very good opportunity for competitiveness.

A SME can have greater flexibility in collaborative work than a large company, an aspect that can be used to achieve greater client proximity. Although this aspect is currently being used in the SME electric machinery sector. The integration and customization needs of these companies are increasing, and classic development schemes based on the definition of initial requirements by the customer tend to be inefficient for both the manufacturer and the client. The client tends to demand hard attainable requirements that must be modified during the design phase (higher engineering costs for the manufacturing company), the manufacturer tends to reduce development costs by trying to reuse previous own designs complying with specifications even if it is not the solution that has the best performance for the client. These schemes based on closed requirements do not take advantage of all the collaborative development capabilities of the manufacturer with the customer. In this sense, a paradigm shift with the manufacturer's relationship with clients is required. Currently, agile development methodologies in the field of SW development, and LEAN Design in the field of the automobile sector have changed the paradigm about the relationship with the client, since the client has become an active stakeholder in the design process.

New products derived from the emerging markets, applications, and new requirements (high efficiency, high comfort, etc.) represent an opportunity for market positioning of SME against large companies. Since these require a quick value proposition either for a new product development more competitive to the new application or for a customization more adapted to the client. In these new applications, the ability to reach the market quickly (short time to market) is critical to achieve a competitive positioning. Making possible that large clients can commit to collaborate with a SME manufacturer.

The product customization, greater integration of equipment, new product development, and short time to market represent greater needs of design resources. In contrast to these needs, today there are new commercial tools for general purpose of design and specific design of electric machines. These tools are based on knowledge (MBSE, optimization algorithms or artificial intelligence AI, etc.). In addition, there are new HW and SW developments that offer current information technologies, to store, process and organize large amounts of information.

Challenges for the new Design Framework:

- The design framework should integrate the client in an agile and efficient way to be an active subject in the design process.
- The design framework will use the maximum capabilities of tools based on knowledge and IT technologies, in order to reduce engineering/validation efforts and the reduction of time to market in the development/customization of new products. In these new tools, design functionalities based on Modular Designs and Design for Manufacturing should be integrated in order to reduce investments and manufacturing costs.

1.3 General framework for Electric Machine Design

In this section, a conceptual framework is defined presenting the different reports, drawings, tools and processes that are associated in the design of electric machines.

1.3.1 Basic documentation defining an Electric Machine Design

Typically, a design of an electric machine is defined by the following documentation.

- **Requirements document (RD)** is a document containing all the requirements to a certain Electric Machine. It is written to allow understanding what a product should do. However, a RD should generally avoid anticipating or defining the electric machine design in order to later allow the interface among designers and engineers to use their expertise to provide the optimal solution to the requirements. In this document, the preferred Key Performance Indicators (KPI), typically efficiency, power density, cost effectiveness, torque quality, mechanical robustness, RAMS indicators, vibro-acustic levels, etc. are implicit or explicitly included. It should be remarked that the maximization of some KPIs implies a reduction of others KPIs. For example, typically an increment of efficiency implies a cost increment. Normally the preferences of the client about the optimal ratio of the KPIs maximization is not included. The lack of definition is considered a source of extra iterations of designs.
- **Design Document (DD)** is a technical document where the product definition of the electric machine is justified. This justification is based on calculations of the different physical phenomena, applying sizing rules, using previous technical solutions for similar design, using a solution that is standardized in benchmarking, etc. A good design document is essential to identify the reasons of a non-adequate performance during the testing phase or to improve the design in a future continuous improvement program.
- **Product definition documentation** (PD): is a list of documents and drawings allowing to define the electrical/mechanical ratings (voltage, current, speed, input/output power levels, etc.) and to manufacture and mount all the pieces of the electric machines. Typically the following information is provided:
    - BOM: Bill of material.
    - Mechanical drawings of each piece including dimension tolerances.
    - Mechanical drawings of the different mountings including dimension tolerances.
    - Electric diagrams of windings.
    - Electrical schemas of the external connections and internal circuits.
    - Extra technical documentation.

It should be remarked that it is very important to define without ambiguity the full design in order to avoid a non-adequate electric machine manufacturing. It is very important to define the admissible tolerances of all geometric dimensions, adjustment mountings, and material properties.

- **Testing Report** (TR): is a technical document where results of tests are presented evaluating the grade of achievements of the performance requirements defined previously in the Requirements Document. A non-compliant result will derive in a revision of the manufacturing process or a revision of the design, needing to return to the design process. Finally, after some design iterations if the design is not able to fulfill the requirements, a new requirement agreement will be negotiated with the client.

Fig. 1-2 shows all the required documentation to fulfill an electric machine design. Traditional designs were based on paper documents. By the introduction of Personal Computers in the companies, the electronic documents were standardized allowing to reduce the time to write and draw the documents, storing all the information and re-using all documents for new designs.

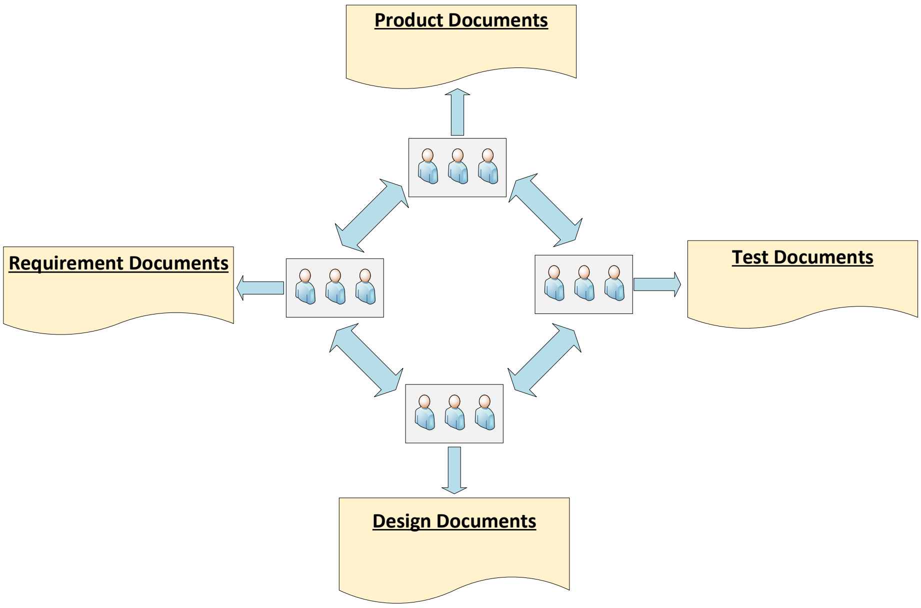


Fig. 1-2. Basic documentation defining an Electric Machine Design.

1.3.2 Advanced tools for the Electric Machine Design.

Nowadays, advanced tools have been developed to support the designer to manage and solve problems associated to the different design tasks.

Next, a brief explanation of the different advanced tools is presented:

- *Advanced tools to define, verify and track requirements:*

In complex electric machines designs different actors participate in defining, checking and refining the requirements. To improve verification and traceability of requirements along the designs, new specialized software has evolved to manage the requirements portfolio (e.g. IBM Rational Doors).

- *Advanced tools to create 3D definitions of the product model*

Nowadays, generalized 3D CAD software programs (Solidwork, CATIA, etc.) have been expanded in the product development companies. These CAD programs allow increasing the productivity of the designer, improving the quality of design and improving communications through documentation.

- *Advanced design tools for a high fidelity computation*

At the same time, specialized CAE (Computer-aided engineering) software packages have also expanded the capabilities of complex computation of physical problems with high fidelity. In the field of Electric Machines, FEM, CFD and Lumped Parameter design tools offer an accurate representation of the mechanical, electromagnetic and thermal behavior. As the obtained fidelity of these tools is high, a big number of different design solutions can be tested and optimized during the design tasks without the need of extra prototypes.

- *Advanced data analysis tools to compare different virtual designs*

Taking into account the computation capacity of these advanced tools, virtual testing results can be obtained. These results do not avoid the need of real testing results, but these results can be previously presented to the client during the design tasks before closing the final design definition. These virtual results allow and obtaining a rapid feedback about which are the performances that the client would like improving.

Fig. 1-3 shows an evolution of the design framework including advanced tools and virtual performance results of different feasible designs. It should be noticed that by using these advanced tools a big number of design variations could be generated and compared. The amount of data generated during the design tasks can be quite high, requiring a data management strategy.

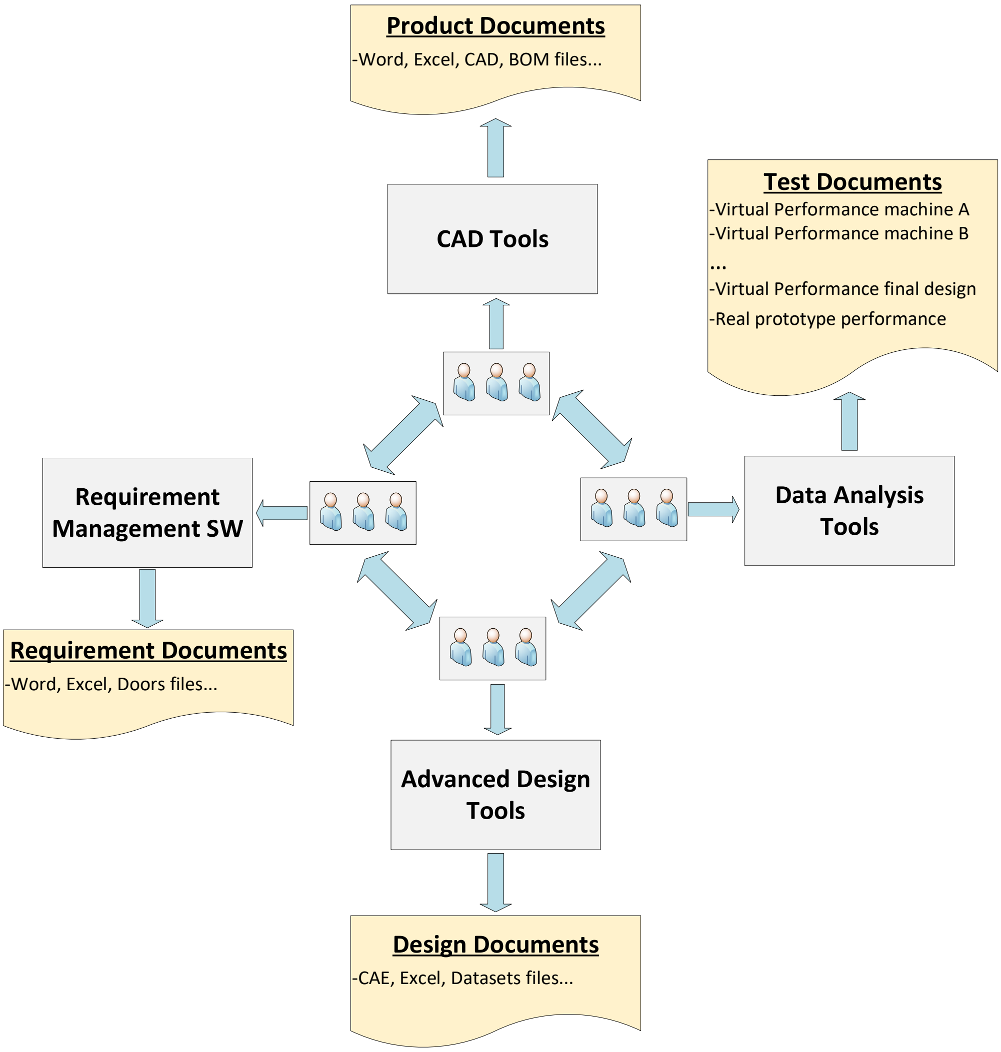


Fig. 1-3. Advanced tools for the Electric Machine Design.

1.3.3 Standardized process for the Electric Machine Design

In a product development environment, it is very important to have powerful design tools, but it is also essential to have standardized processes to carry out the sequences of full design tasks. In this sense, according to the description of process standardization coming from Morgan and Liker's 2006 book on the Toyota Product Development System [1], the company Toyota claims that their highly standardized processes for product development allow creating highly stable and predictable outcomes (with both quality and timing) in an unpredictable environment. This necessity of standardized processes is directly applicable to the electric machine development environment. It should be remarked that the design tasks for electric machines are complex related to a parallel multi-physics design (electromagnetic, thermal, mechanical, etc.) where several engineers could work at the same time on parallel processes and whose results have to be integrated. A time delay in a task-result can stop the workload of other tasks. Even worst, a computation error in one design domain (e.g. electromagnetic design) can imply serious consequences in a re-design of the others domains (e.g. thermal, mechanical). Then, standardized processes are needed to assure highly quality standards in each individual outcome and to manage the iterations among the different sub-processes without uncertainties.

In order to support the standardized definition of the process in complex systems without ambiguities, Model Based Systems Engineering (MBSE) frameworks are being used. According to INCOSE (International Council in Systems Engineering), MBSE is “the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later lifecycle phases” [2]. Different standardized modelling languages can be used to support the modelling diagrams that requires MBSE. For instance, the SysML language that consists of different diagrams standard based on behavior, structure and requirement representation of engineering systems.

Fig. 1-4 includes the processes to obtain the documentation in the design tasks. It should be noticed that in these processes the different actors (designers, clients, others) have to follow the specific flowchart that it is involved in each moment. A SW program can support this flowchart but it is not strictly necessary. The important issue is that each outcome or problems were registered (including the design data) in the global data management of the full macro-process. One objection of designers to work in a structured flowchart could be that their creativity is reduced inside the constraints of the process. To avoid this issue, it is necessary to assure that each micro-process includes an adequate degree of freedom of the design variables to cope with the optimization goals required by the designers.

At the same time, during the design process the designers can derived new designs concepts that due to their uncertainty cannot be directly included in the structured process associate to a current product development project. To avoid the loss of creativity, companies should include internal open innovations programs to register the obtained new ideas during the process, to research and develop these ideas and to improve future design processes with better standard micro-processes and new design solutions.

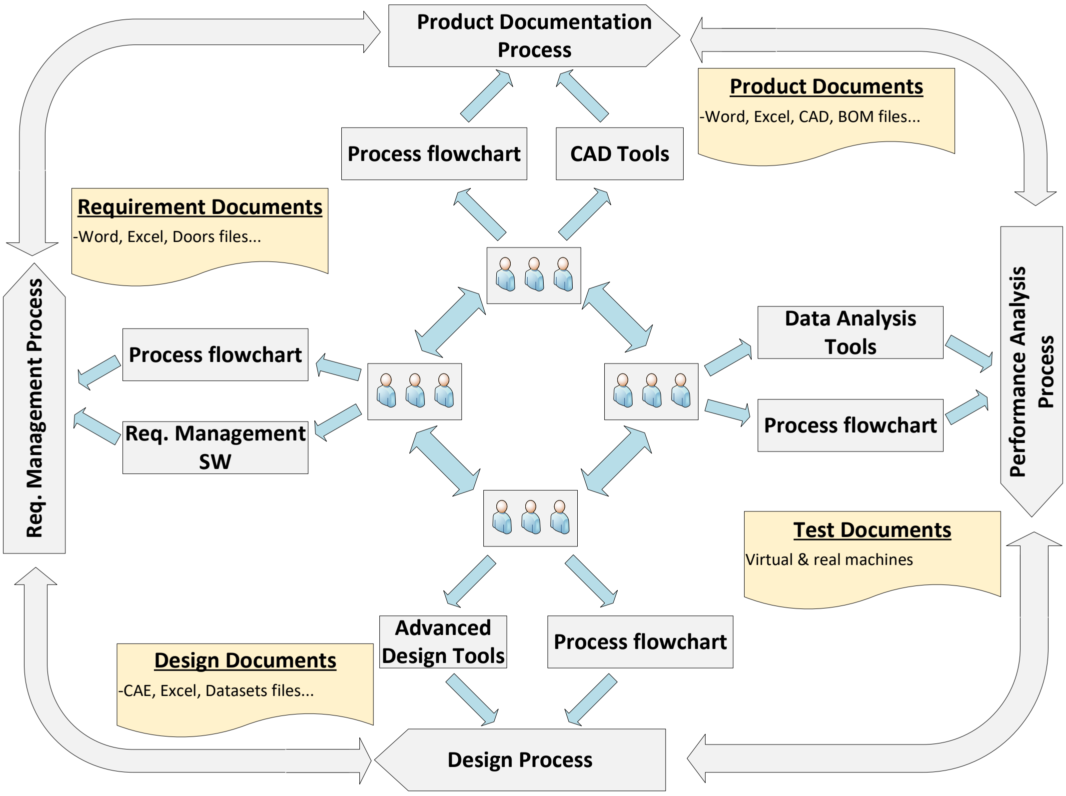


Fig. 1-4. Standardized process for the Electric Machine Design.

1.3.4 KBE system for the Electric Machine Design

It should be remarked that the standardization of some creative tasks during the micro-processes is not straightforward. For example, to solve an unpredicted problem during a design process, it is common that the designer remembers a similar problem that happened in the past, recovering in their mind a good solution for this new problem. Thus, the knowledge of the designer cannot be substitute by a standardized process. However, at the same time, this knowledge of each designer is hidden to the rest of the organization and it is only useful when the designer is involved in a specific problem. In addition, this knowledge is lost when the designer is retired or takes another activity. In order to capture and reuse the knowledge of the organizations, in the scientific literature Knowledge Based Engineering frameworks have been proposed.

Knowledge based engineering (KBE) is a branch of knowledge-based systems applied to the domains of design, manufacturing and production. Usually applied for mass customization, its used have been helpful to reduce time in the design process by avoiding repetitive tasks for the designer and also accumulates information and makes it available for reuse [3].

KBE solutions contain expert knowledge about how, when, and what needs to be done, to be identified, gathered, and further processed by a computer system allowing its easier application in new projects. A formal description of rules applied by the design engineers helps to process standardization and allows automation of repeatable tasks in the design process [4]–[6]. Application of solutions based on knowledge description at an early stage of product development also decreases costs of the whole project and facilitates selection of the best solution [7]. Building and implementing a KBE system in the design process of variant products can be a way in which a company can realize assumptions of the mass customization strategy [4], [8].

The most important reason to consider when building a KBE system is, above all, the possibility of rationalization of the design process. It is estimated that approximately 80% of design time is spent on routine tasks [9], [10]. Acceleration of routine tasks can significantly help optimize the whole product lifecycle and cause certain savings. This is why a basic advantage of using KBE solutions is the possibility of automation of repeatable tasks [11], simultaneously influencing capabilities of realization of creative work [6]. Fig. 1-5 highlights this improvement and a review of KBE applications with its achievements can be read at [3].

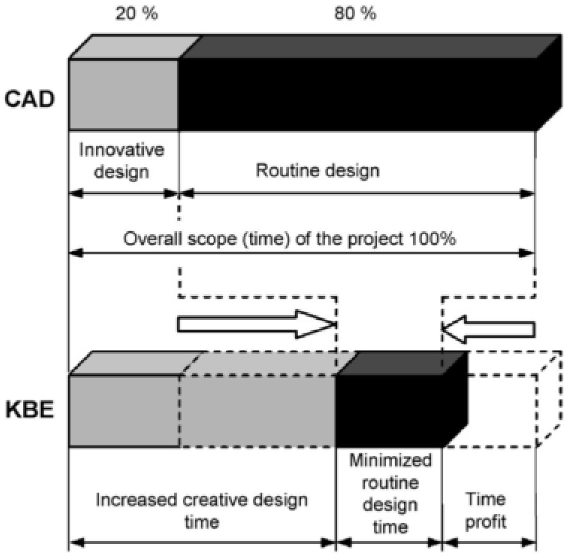


Fig. 1-5. KBE routine design time reduction [6] p. 678.

Although, there are methodologies (e.g. MOKA, KNOMAD) to carry out the process to capture, formalize and package the knowledge, there is not a standard implementation of a KBE system.

To understand the framework, the following concepts definition are used throughout this work.

KBE application: *a specific-purpose software component developed with integrated knowledge to solve knowledge intensive repetitive problems. It is a result of the KBE system framework.*

KBE system: *a system that consists of a suite of KBE applications and their integration with other entities (software components, actors, etc.) to pursue a goal for the product development process. It is a result of the KBE system framework.*

KBE system framework: *the structure underlying the creation of KBE applications and KBE systems that can be selectively adapted to suit a particular need for a product development process.*

Fig. 1-6 shows a conceptual KBE system framework for the design of electric machines. In this figure, a KBE application is defined for each specific process, but different configurations can be implemented grouping various processes.

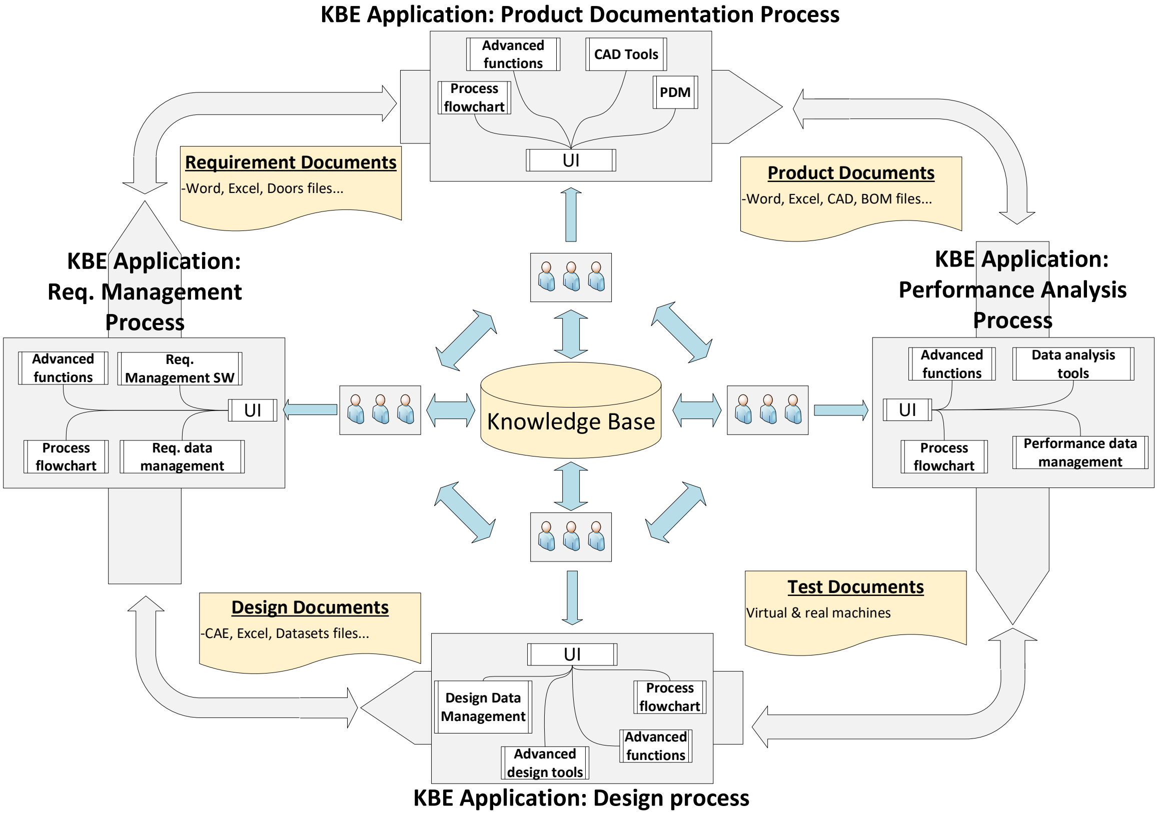


Fig. 1-6. KBE System for the Electric Machine Design.

This conceptual KBE System framework has the following basic components:

- **Knowledge Base**: It has knowledge about the product & process modelling of the full electric machine development process. This knowledge could be represented through standardized informal and formal reports, diagrams, and models. For knowledge formalization a standardized language is recommended to avoid ambiguity in the knowledge representation.
- **KBE application**: To elaborate more about its definition, this SW program is developed using the formalized knowledge obtained from the Knowledge Base. In the KBE application, different functionalities can be implemented. Next, as example some interesting functionalities are presented:
    - **UI (User Interface).** An easy and flexible UI could allow the designer to manage all the functionalities and to know by a simple visualization the current state of the design. It is important that the UI can offer an adapted scenario for each user role and design stage.

- **Process Flowchart.** This functionality allows the designer to go automatically along the flowchart. The SW will indicate in each moment: the role and to do task by the designer as well as the conditions to continue/return along the flowchart.
- **Advanced design tools.** The KBE application should allow managing the advanced design tools of the company. For this purpose, middleware components would be interesting to be implemented for automatic iteration between the KBE application and commercial SW programs. Anyway, more than an automatic iteration between programs, the most important feature is the capability of the KBE application to apply an intelligent interpretation and management of the input/output data from the different design tools. For this purpose, standard metadata interchange protocols can be used, such as XML Metadata Interchange (XMI®).
- **Historic data management.** This functionality allows registering all the product data and the process milestones and outcomes. At the same time this functionality allows accessing the required historic data of the company about the products and processes.
- **Advanced functions:** Several advanced functions can be included to help the designer to reduce the routine task time in order to increase his effort in creativity tasks. For instance, the following functions can be included:
  - **Functions for automating** the workload inside the micro-processes.
  - **Intelligent searching functions** to recover from the historic data the most similar design solutions or problems to the current design.
  - **Intelligent support functions** to give extra information to the designer during the task development. This help can be used to present the specific knowledge about the problem in which the designer is involved.

1.4 PhD thesis framework

Taking into account the SWOT analysis (Section 1.2.2) in the development of electric machines the following competitive **drivers** are concluded:

- D1. Complete intelligent use of the present and future **knowledge of the company** derived from people, tools, processes, etc.
- D2. **Product customization** for the application and for the **client** needs.
- D3. **Cost-effective design and manufacturing** solutions.
- D4. Continuous **Technological Innovation** allowing developing new innovative and more competitive products taking advantage of new materials and new manufacturing processes

Fig. 1-7 shows the generic vision of including the KBE system into the global electric machine development and life cycle processes. It should be noticed that thank you to Big Data and IoT (Internet of Things) technologies, the real data of electric machines (working operations, health states, service problems, and others) could be accessible for the electric machine manufacturer offering new opportunities for the product customization and improvement. In addition, Big Data analysis could allow developing an accurate correlation between the health state of the machine during its life cycle VS the traceability of the materials and manufacturing processes used in each individual machine. The knowledge obtained from these correlations will offer new opportunities to improve the quality/cost ratio of the materials and manufacturing processes. Considering the design processes, it is important that the KBE system for design have to allow the following features:

- F1. **Integrate Innovation** in the design processes.
- F2. Industrial design based on **Modular Designs** methodologies and **Design for Manufacturing** concepts
- F3. **Integrate the client in the product customization** not only in the initial requirement phase, but also along all the design process to obtain the most suitable relationship between the different KPIs from the direct perspective of the client.
- F4. **Industrial product design** with **short time-to-market** assuring qualification of requirements.

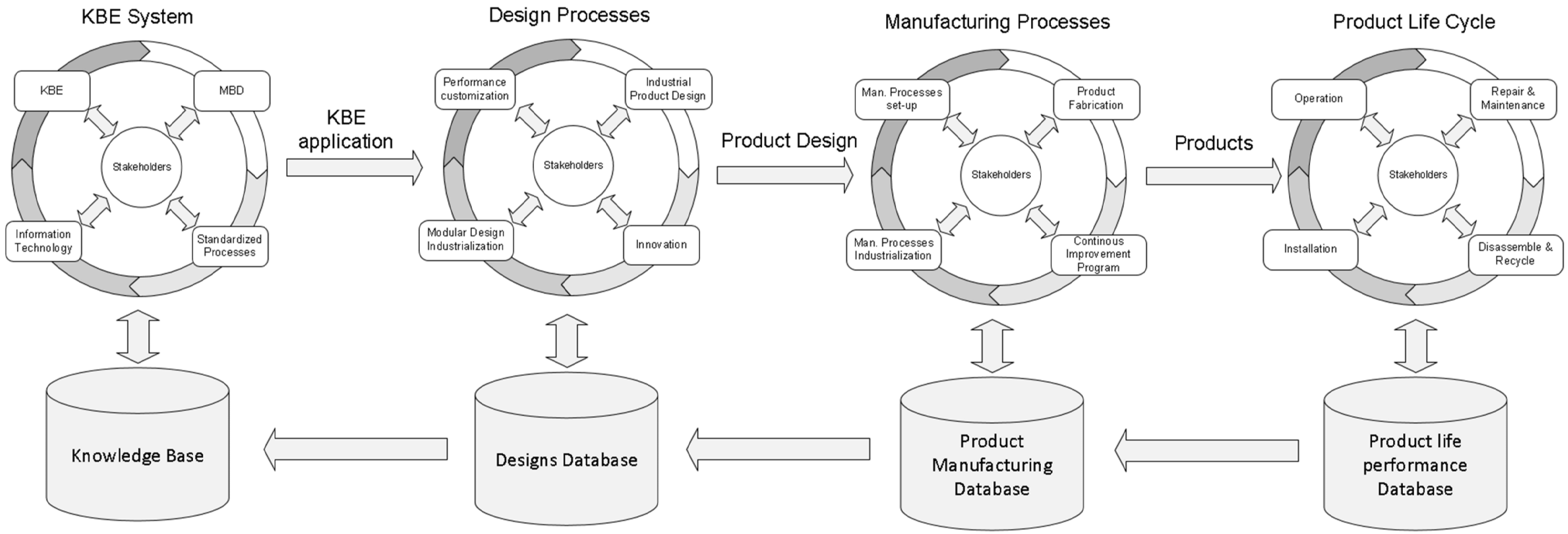


Fig. 1-7. KBE System into the global development and life process of electric machines.

In order to achieve these **drivers** and **features**, in this thesis it is proposed to develop a new KBE system taking advantages of the following engineering domains:

- E1. **Knowledge Based Engineering** to capture, formalize the knowledge and to develop advanced design assistance SW.
- E2. **Model Based Design** (MBD) to model the different physical phenomena involved in the electric machine allowing a better design optimization and a virtual development framework.
- E3. **Standardized processes** taking into account the different development maturity levels of the technology inside the company (concept level, components industrialization level, industrial product level).
- E4. **Information Technologies** to register, manage, and analyze all recorded data/information of the company, allowing the automation and easy use of this data/information by the different design processes.

Next figure resumes the key points of the drivers, features and engineering domains.

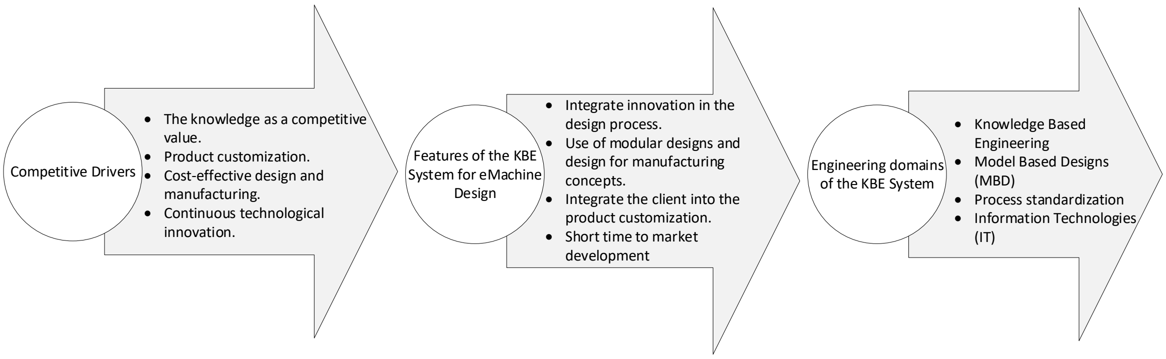


Fig. 1-8. Key points for the development of a new KBE System.

Regarding the Model Based Design tools, during the last 15 years technological advancement has been produced in the design tools for the development of electric machines. The analytical and numerical tools (FEM, CFD) initially developed and validated in research studies, lately have been transferred to the CAE commercial programs, allowing being accessible for any electric machine manufacturer.

The latest technological developments related to the IoT, Big Data and Cloud Computation have derived in the availability in the market of very powerful generic IT platforms allowing managing a huge amount of data with very high computing capacity. These generic IT platforms require less intensive financial investments and they allow an easy and fast implementation.

Regarding the development of electric machines, scientific literature about the standardization of the processes integrating several TRLs (Technology Readiness Levels) and higher client integration perspective have been short. It is considered that this is due to:

- A major interest in the scientific community to develop more innovative electric machines, than to improve the process to customize the current technologies to the industrial demand cases.
- The companies that have developed their own design processes consider this information as a confidential issue.

At the same time, KBE systems have not had a clear framework to be directly used for electric machine design purposes. Noting that the derived KBE applications should integrate the advanced design tools in an easy and flexible way as well as the evolution of commercial design tools is very fast. Therefore, it is necessary to implement KBE applications with a very flexible architecture in order to be adapted to new features coming from the commercial CAE programs. At the same time, the capture of the knowledge from specialized designers that are experts in one engineering domain (electromagnetic, thermal, mechanical, insulation) is not a straightforward issue, requiring new methodological frameworks and adapted implementation tools.

Taking into account these last considerations about standard processes and KBE systems, the scope of this thesis is defined as:

Scope of the thesis focused on:

- The standardization of the design processes of electric machines, integrating various TRLs and higher client integration perspective.
- The development of a KBE system framework for the design of electric machines including the formalization for Knowledge Base generation and the design of flexible architectures to adapt the KBE systems to the new technological improvements that will come from Advanced Design Tools and from IT platforms.

1.5 State of the art

In the state of art, the two main lines of the scope of the thesis are studied: Knowledge Based Engineering (KBE) and Standard Processes for the development of electric machines.

1.5.1 Knowledge Based Engineering (KBE)

1.5.1.1 General methodologies and tools

KBE has had broad applications and success in mechanical engineering specially in the automotive and aerospace industries [6], [12]–[20]. Usually, KBE is related to where computer aided design meets artificial intelligence for the automation of repetitive tasks in the design process, but lately the applications has extended its definition to the integration of other software tools also allowing the improvement of collaborative designs which have lately led to the integration with PLM [4], [7], [8], [21]–[23].

After researching about the topic, it can be concluded that there is not a clear-cut definition of KBE, researchers define it according to their perspective and application, nevertheless, there is a common denominator to automate designers' repetitive tasks using engineering knowledge in order to reduce time and costs of the product development. KBS started to penetrate the market around the 70s, however, not with the expected impact for the design process in industry including the automotive and aerospace areas where its use is common today. The reasons of this failure in industry during early development were excessive costs of software, lack of methodologies, no examples in industry applications, localized implementations, high learning curve and low design expectations [12], [23]. The first KBE tool in the market called ICAD was developed by the AI laboratories of MIT and Computervision now PTC; Lisp programming language has been the origin of new languages such as GDL (General-purpose Declarative Language) of Genworks, AML (Adaptive Modeling Language) of Tecnosoft, IDL (ICAD design language) of MIT and Intent! of Siemens [12].

Nowadays, due to cost on hardware and software are more accessible, KBE can be rapidly implemented in any needed stage in the design process. KBE is useful for capturing human design process, supporting automation of tasks, integrating other software, designing products involving high learning curves [10].

KBE delivers the documentation related to the developed product model by transforming the knowledge (rules, standards, materials selection, and others) into the desired model which must satisfy the input requirements, this process is illustrated in Fig. 1-9.

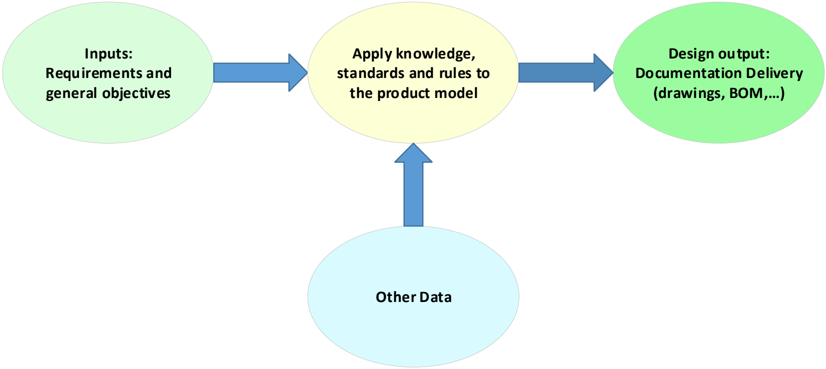


Fig. 1-9. KBE process. Adapted from [3], [10].

Verhagen et al. [11] and Jayakiran et al. [3] identifies some shortcomings of KBE. These are a lack of formal methodologies or structural framework, the ‘black-box’ application tendency of development, lack of knowledge reuse, failure to include quantity assessment such as costs. Furthermore, no procedure to identify and justify process suitable for KBE usage, building and implementation takes a lot of time and hardware and specialized skills, and finally because of its localized implementation KBE could not register its impact in complete product development process.

Problems of gathering, representing and applying the knowledge in KBE computer systems are dealt with the domain of knowledge engineering, which is known as a branch related to the creation of knowledge bases and use of semantic technologies to process knowledge through computer systems [24]. Knowledge should be recorded in a knowledge base for its effective application in the design process. To do so activities such as identification, acquisition, representation and analysis of knowledge are carried out. Furthermore, the creation of databases and repositories of knowledge takes place in order to have access to the knowledge [24].

These apparently obvious activities actually present a number of huge challenges for engineers, increasing the cost and development time of these KBE systems [11]. This happens in spite of available methodologies for building KBE systems, which have been present for years and are aimed at ordering knowledge-processing procedures. Examples of these are methodologies such as DEKLARE [25], KOMPRESSA [26], KCM [27], MOKA [28], and KNOMAD [29]. A key research challenge is formed by the need for better support and adherence to KBE methodologies in KBE development projects. To achieve this, a number of issues should be researched, the foremost of which is closing the ‘‘technology gap’’ regarding the lack of tools and technologies to support cost-effective KBE development. Other important research issues include achieving transparency of KBE systems, enabling knowledge sourcing and reuse through data mastering and standard development, enriching the semantics of knowledge models and ensuring traceability for implemented KBE solutions, reporting KBE success metrics, and providing an assessment framework for KBE opportunities [11].

1.5.1.2 A review of KBE systems for electric machines

The importance of integrating new technologies to support the development process as well as knowledge retention and transfer in the electric machine industry has been identified also by other authors [30]–[32]. All agree that the motor design and production has increased in complexity lately. Therefore, there is a need of new methods to achieve demanding requirements from policies and regulations as well as applications for instance the transportation industry. This thesis started in 2016 with its first publication in 2017 showing the concept of what this research attempted to obtain. Next, literature regarding KBS for electric machines are now summarized.

Mayr et al. [30] have recently published a concept toward the creation of a knowledge based design methodology for managing the complexity of the integrated product and process development of electric motors. Their research is still ongoing and they aspire to create a web-based software implementation. It seems interesting the way they want to integrate manufacturing processes.

Karnavas et al. [31] propose a knowledge based software architecture scheme implemented for the design of electric motors. They structure the KBS in different layers consisting of knowledge sources, and depending the layer reasoning modules obtain or filter design alternatives towards an optimal solution.

Favi et al. [32] presents a web-based software platform called EROD (Energy Reduction Oriented Design) to configure and simulate customized energy efficient motors. The platform core is a knowledge based system. The design methodology is oriented towards eco-design, the knowledge based system guides the user in entering input parameters and configuring the design solutions based in the know-how previously formalized in terms of rules. It also integrates design and specific tools including cost analysis. It generates automatically technical documentation. This work is more focused on the specific implementation, more than on developing a general Knowledge Based System.

Cross et al. [33] presents an expert system for the electromagnetic design, integrated enclosure specification and commutation test diagnosis of DC motors developed in new created shell with turbo-prolog language. The system is divided into three different expert systems one for each of the solutions aforementioned. Backward chaining inference technique is used, although a combination of backward and forward chaining is seldom used too.

Landy etal. [34] presents an expert system able to lead an amateur designer for the design of 3-phase induction motor. A new designer is provided with options in the whole process in order to help him make accurate decisions and also gives information about every decision that is about to be made. According to the authors of the paper, the system showed outstanding results for educational applications. The knowledge base is described and the motor is designed in 13 steps.

Miljavec [35] presents the possibility of building an experts system using artificial neural network (ANN) for the design of a synchronous reluctance rotor. The main goal of the paper is to know how the radial magnetic flux density is distributed along the airgap. The feed-forward artificial neural network is used and satisfactory results were obtained using the “trainlm” function. Thus concluding that the implementation of ANN is promising for building ES for reluctance motor design.

Zhou et al. [36] presents the use of the unified modeling language (UML) to represent the design of an expert system for the selection choice of small and medium electric motor of sewing machines. It mentions the criteria for stablishing the rule base and fact base databases and it uses Jess as a shell with Java. UML is an object-oriented visual standard modeling language that uses graphics symbols to express the objects and their relation in the system.

Scutaru et al. [37] presents a software written in visual basic that is able to optimize or redesign three phase induction motors. The program obtains information of existing magnetic circuits of previously designed motors from a database. The program main menu includes three components which are the general data component, in this form the selection of the materials is possible based on the desired characteristics (steels sheets, conductors, insulation, etc.). The next component is the computational variants where geometry dimensions are set. Finally, the gauging component where correction factor can be applied to iron losses, stator resistance, magnetizing current and stator slots filling factor.

Jabbar and Yeo [38] presents a software based in MS Excel with macros for the electrical and magnetic design of brushless motors for low power range.

Laugis and Vodovozov [39] propose a software tool that generates and edit structure query language sentences to extract information from databases in order to suit multitude sets of components for a particular application.

Xie Wei et al. [40] present an expert system for the design of permanent magnet synchronous motor. The expert system is developed using Visual-prolog. A database has been develop with different design schemes collected from manufacturers of permanent magnet motors of 20 power ratings and different magnetic materials. The initial design scheme is taken with the analogism method from this database. It uses the rule skeleton as the level which describes the relations between all nodes in the inference network and the rule body as the level that describe the fixing value method of nodes respectively. The inference network is composed of three node sets, the top node set by the specification variables, the bottom node set by the performance variables and the middle node set by the variables that may vary to achieve the best machine. A design example is presented satisfying the required specifications, therefore, validating the expert system.

Lowther [41] presents the order on how the knowledge regarding electric motors should be structured. First starts stating an electrical machine has a define structure given by the intended application and the laws of physics. The structure is described in a hierarchical form, an example of levels is given starting from the whole concept of the electrical machine breaking down more levels of more detailed parts as stator and rotor, and these also broke down into other levels and so on, until the desire complexity of components are defined. This frame based approach often referred to as Deep Knowledge. It suggests to use the case based approach where the initial design is chosen as an existing design which most closely matches the current requirements. However, is also important to bear in mind, that one of the limitations of this approach is the lack in innovation. Finally, to have a consistent communication medium between different areas a Blackboard System is suggested as a common database supporting knowledge and rule based systems to provide a mechanism for opportunistic design.

From the literature review, it can be concluded that to date the implementations of KBE design systems for electric machines developments have been derived taking into account the specific problem. This approach reduces the cost-effectiveness of the developing KBE system. Thus, there is a lack of methodologies to develop KBE systems. At the same time, few applications have been developed for electric motors. These applications have been developed as a research proof of concept, without being commercially used. Therefore, there is the need to propose flexible architectures frameworks to allow a higher number of cost-effective KBE implementations with a final oriented use by electric machine manufacturers.

This research proposes to create a knowledge based engineering system framework that leads to the creation of KBE applications for the development of electric machines. Additionally, that allows integrating top-of-the-line tools for electrical machine design and the know-how acquired by the company, endowed with flexibility to contemplate previous designs and to integrate feedback from the electrical machines experts' about the results obtained from the KBE applications. Thus, the KBE system will have the features such as automation of repetitive processes, integration of multiphysics and analytical design tools, data management and intelligent algorithms.

1.5.2 Design processes for electric machines

This section will review some representative studies about the design processes of electric machines. Although a vast literature regarding electric machines exist, these literature is more focus on new scientific developments from academic perspectives, and few attempts are presented with a systematic approach about the full design processes of the electric machines. It is true that each manufacturing company has developed its own full design processes according to their needs and their historic R&D developments, but as these processes are considered as a confidential know-how, they have not been published.

Deshpande in [42] covers the design of electric machines, the text goes from basic principles on design up to specific procedures to design induction, synchronous, dc machines and other machines. It also presents a chapter with design synthesis using CAD systems. Other texts are more specialized in a specific technology [43]–[45], for instance [43] presents the theory to design brushless permanent magnet motors in a practical way with the gained experience of the authors in these type of machines, also considering software-based techniques. While [45] emphasize on induction motors.

Pyrhonen et al. in [46] covers the design of electric machines focusing on the electromagnetic design of induction machines although with few variations in the process, also suggested in the text, it can be applied to other motors. The design process suggested by [46] is shown in Fig. 1-10.

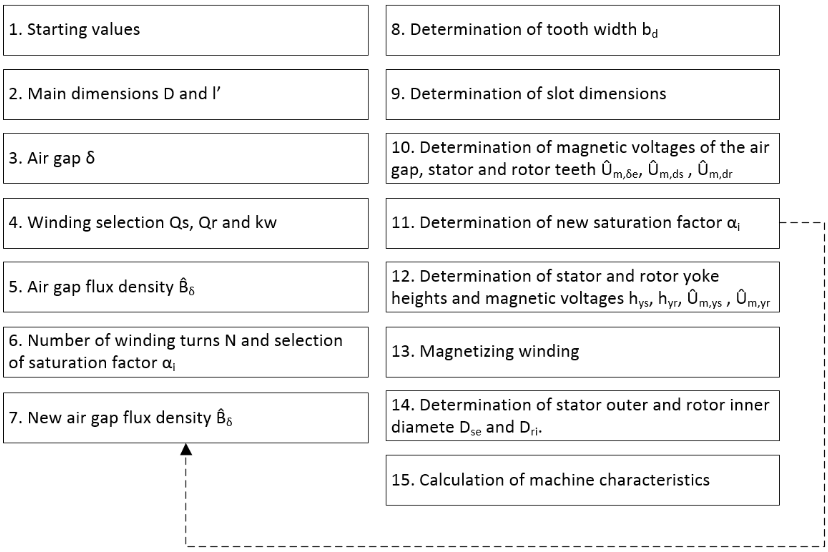


Fig. 1-10. Pyrhonen design process for the induction motor [46].

When creating an electric machine not only electromagnetics factors are involved but also mechanical, materials and thermal features among others should be considered. Most literature centers their study only on the electromagnetic design of electric machines, neglecting in some way other important domains. Tong in [47] mentions aspects that should be taken into account in the motor design process, these are illustrated in Fig. 1-11. He also encloses the mechanical design features of electrical machines. The design process proposed by Tong [47] can be seen in Fig. 1-11.

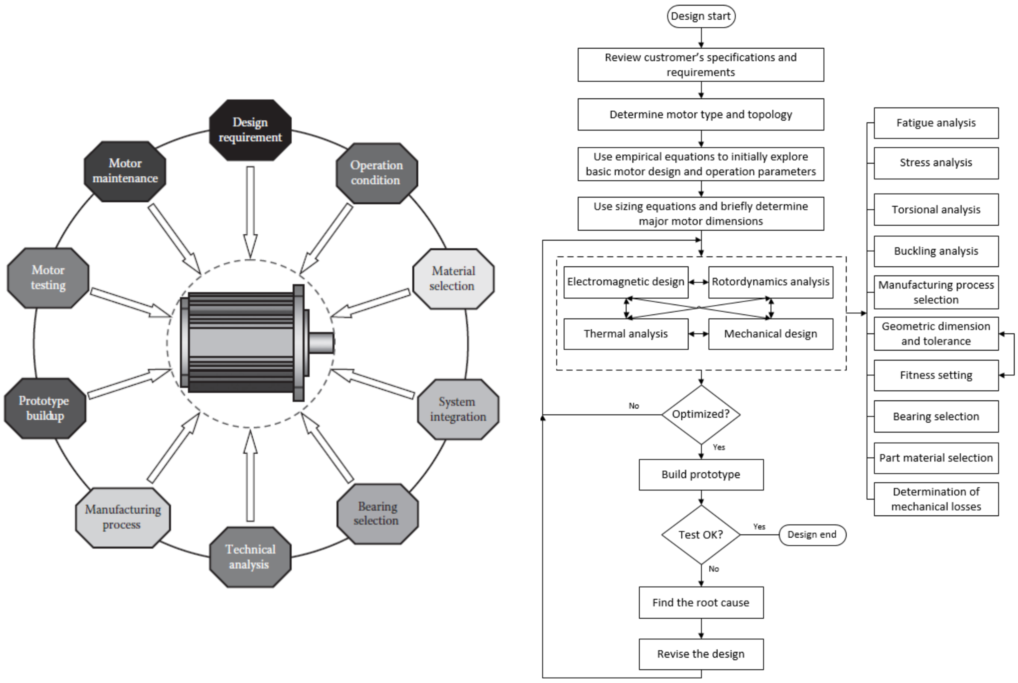


Fig. 1-11. Design process proposed by Tong in [47] p. 69, 70.

In summary, every design methodology of electric machines have in common as first step the definition of requirements of the application i.e. specifications. After this step, the selection of the motor type and topology is usually followed and continued by the dimensioning of the machine. Here electromagnetic aspects are considered i.e. design of the magnetic circuit with the proper materials selection [48], [49] , and the electrical design as stator winding configuration, number of poles, rotor windings (if required). In literature, most contemplate in this phase the thermal features on the design and few others the mechanical aspects [47], [50], [51].

The next step is the design optimization, the use of analytical tools and FEM analysis is the common denominator with designers [32], [52]–[58]. In this part the iterative process is usual, nowadays, the trend is the implementation of multi-objective optimization algorithms as tools to achieve time reduction on the selection of the best machine [59]–[64]. Due to the aid of these algorithms, designers are able to focus on more creative factors on the design and also a reduction in design costs in the life cycle of the motor is attained. Finally, a prototype is built subject to performance tests for final validation results to check its feasibility as well as the approval of the design process.

In conclusion, regarding the electric machine development process different scientific publications propose methodologies based on the designer know-how and experience in the field. Moreover, these methodologies even vary depending the types of machines to be designed although with some effort most can be adapted to other types of motors. Another issue is the need of integration of all domains involved (e.g. mechanical, electromagnetics and thermal) as well as the tools to enhance the development process to achieve a feasible design in a shorter period.

Taking into account the full design framework to be developed in this thesis, it is concluded that processes and design tools of the literature can be used as micro-level processes inside a domain-specific design task. These micro-level design processes could be specific for each electric machine type and/or each application need. At the same time, it is envisaged to use a macro-level design framework to maintain a common process structure to all designs of the company.

1.6 Motivation

Regarding the scope of this thesis, “A KBE system framework for the development of electric machines including a standardization of design processes at several TRL levels”, the State of the Art has the following open research issues:

- *There is a lack of a common framework for the standardization of the Macro-level process at various TRL levels.*

  The integration of the client perspective is only done during the initial specification stage. Product customization requires a higher integration of the client during all design phases. At the same time, it is necessary their integration in the complete design process of the different development levels of technology and products, including: a continuous innovation, a cost-effective technology development and a short-time customization of a new product.
- *There is the need of KBE methodologies & technologies adapted for the electric machine domain in order to support a cost-effective KBE system development.*

At this point, the following relevant challenges have been identified in this thesis.

a) To develop a standardized methodology to capture and formalized the knowledge for its implementation specifically in the electric machine domain.

b) To generate a generic Knowledge Base of products and processes through modelling that could be applicable in the majority of electric machine developments.

c) To propose SW architectures, commercial SW platforms, tools and procedures to develop a specific KBE system in shorter time and with lower implementation cost.

d) To increase the cases for KBE applications implementation in real industrial scenarios. Due to the need of high level acceptance of the designers and clients, with respect to the SW developments is better to use a Minimum Viable Product (MVP) approach in order to obtain faster the stakeholders’ feedback and to be able to develop a SW more adjusted to real needs of designers and clients.

1.7 Objectives

The main objective of this thesis is to define a knowledge-based engineering (KBE) system framework for the development of electric machines. To accomplish the main objective the following partial objectives are defined.

1. To propose a complete range of standardized processes to cope with the development phases at different levels of technological maturity.
2. Using a generic KBE methodology, to develop an adapted derivation for the electric machine domain to capture and formalize the knowledge.
3. To create an initial generic Knowledge Base of products and processes that could be reused in the development of electric machines.
4. To propose a SW architecture based on commercial SW, advanced design tools, and a customized and flexible KBE application to accomplish the real needs of a manufacturer.
5. To develop an MVP concept for a KBE application. A KBE application is proposed to customize the final industrial traction motor for a given elevator application system.

1.8 Outline of the thesis

The structure of the thesis is summarized below.

- Chapter 2: Standardized Processes for the Development of eMachines

This chapter introduces a conceptual standardized macro-level framework for the development process of electric machines. The purpose of the framework is to offer a systematic approach flexible enough to be adapted for any type of motor development project. It merges the VDI-2221 and VDI-2206 guidelines to offer a system for the development of eMachines describing in a high-level each macro-process. In addition, a requirement engineering framework is proposed.

- Chapter 3: Knowledge Base Generation

This chapter introduce MOKA methodology. An adaptation and extension of this methodology has been derived to fit the development process of eMachines. A new process model knowledge represention is suggested. In addition a new micro-process metamodel is presented.

- Chapter 4: KBE system framework

This chapter presents the KBE system framework for the development of electric machines. The framework synthesizes the aforementioned chapters providing a complete description of the issues, processes, actors and tools involved to create a successful KBE system. The so-called KBV2-model is shown, which consist of the knowledge base generation process and the eMachine standardized macro-level framework. In addition, a KBES SW architecture is proposed.

- Chapter 5: Micro-level Implementation

This chapter provides two examples of implementation of the KBES framework for the micro-level case. Two micro-level KBE applications are developed to show substantiation of the process proposed in the KBE system framework. Both implementation provide support to designers to compute complex operating points for brushless AC motors.

- Chapter 6: Macro-level Implementation

This chapter provides a practical macro-level implementation example of the KBE system framework. A KBE system is developed by integrating current technologies used to enhance the product development process. The implementation is carried out for the Industrial V-cycle regarding the design of electric motors for elevator systems.

- Chapter 7: Conclusions

This chapter provides the conclusion of the thesis.

1.9 Publications

Journals included in Journal Citations Reports (JCR)

- C.A. Rivera, J. Poza, G. Ugalde, G. Almandoz, A requirement engineering framework for electric motors development, Appl. Sci. 8 (2018). doi:10.3390/app8122392.

IF (2018) = 2.217 (Q2) in "Applied Physics"

Published in section *"Energy"* of *Applied Sciences.* This paper presents a requirement engineering framework to support small-medium electric motors designers/manufacturers with the development of their product. The framework identifies the stakeholders and the tasks that they should undertake to finish a successful requirements specification stage. The framework is made from the designer/manufacturer's perspective and it emphasizes the derivation of specialized requirements (lower-level). The result of the framework is well-defined requirements that form the design requirements specification of the motor that leads to the beginning of the design stage.

- C. A. Rivera, J. Poza, G. Ugalde, G. Almandoz, Field Weakening Characteristics Computed with FEM-Coupled Algorithms for Brushless AC Motors, Energies. 11 (2018) 1288. doi:10.3390/en11051288.

IF (2018) = 2.707 (Q3) in "Energy & Fuels"

Published in section *"Electrical Power and Energy System"* of *Energies*. This paper presents nine algorithms as an alternative to compute with iterative methods operation points that cannot be targeted directly on a FEM tool. The algorithms must be coupled to the FEM tool and can compute complex points such as the characteristic current and modes of operations limits within acceptable range of error and times of execution for practical purposes. Validation of the algorithms using Jython is presented with results for the three types of brushless motors (non-salient, interior permanent magnet and reluctance motor).

Conferences

- C.A. Rivera, J. Poza, G. Ugalde, G. Almandoz, A Knowledge Based System architecture to manage and automate the electrical machine design process, In 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM); IEEE, 2017; doi:10.1109/ECMSM.2017.7945875.

This paper introduces a concept of a new Knowledge-Based System (KBS) model for electrical machine design. The model so-called GSMWV2 (initial concept of this thesis, then changed to KBV2) stands for the steps of Gather, Standardized and Model the knowledge with the Wrapper process and the two Verein Deutscher Ingenieure (VDI) methodologies integration VDI-2221 and

VDI-2206. It comprises the electrical machine development methodology and the knowledge integration process. This paper conceptually describes this model and the integration of current industrial technologies such as Model-Based Systems Engineering (MBSE), Product Lifecycle Management (PLM), Product Data Management (PDM), multiphysics and analytic tools co-simulation with KBS tools. This architecture will be able to manage and automate tasks in the development process of electrical machines with emphasis on the design process

.

Chapter 2

STANDARDIZED PROCESSES FOR THE DEVELOPMENT OF EMACHINES

This chapter introduces a standardized macro-level framework proposal for the development process of electric machines. The aim of the macro-processes is to offer a systematic approach flexible enough to be adapted for any type of electric machine development project. It uses the VDI-2221 and VDI-2206 philosophy by merging both guidelines into one. Adapting these to offer standardized macro-level framework for the development of electric machines.

This chapter is structured as follows. Section 2.1 gives a brief introduction of the chapter. Section 2.2 reviews systems engineering as a discipline that can represent processes without ambiguity. In section 2.3, the standardized macro-level framework is introduced. Section 2.4 presents a requirement engineering framework proposal for the development of electric machines as an extension of the specification stage of the framework. Finally, section 2.5 shows a brief discussion of the chapter.

2.1 Introduction

The purpose of this chapter is to propose standardized macro-processes for the development of electric machines with the aim of offering a systematic approach flexible enough to be adapted for any type of motor development project. The proposed macro-level framework consists of three macro-level V-cycles, the concept, component and industrial V-cycles. Each for particular objectives in the development process. Within these V-cycles stages are described in order to end up with a successful validation of a prototype.

Inside the V-cycles special attention to the specification stage will be included in this chapter. The specification stage of an electric machine is a critical part of its development. If this stage is properly addressed, then future not desired redesigns in the development process can be avoided. This chapter presents a requirement-engineering framework to support small-medium electric machine designers/manufacturers with the development of their product. The framework identifies the stakeholders and the tasks that they should undertake to finish a successful requirements specification stage. The requirement framework is made from the designer/manufacturer's perspective and it emphasizes the derivation of specialized requirements (lower level). The result of the framework is well-defined requirements that form the design requirements specification of the machine that leads to the beginning of the design stage.

2.2 Systems engineering

During the last years, in order to support the standardized definition of processes in complex systems without ambiguities, systems engineering has evolved.

The International Council in Systems Engineering (INCOSE) in its handbook [65] p. 7 defines SE (systems engineering) as a perspective, a process and a profession:

As a perspective: “SE is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect”.

As a process: “SE is an iterative process of top-down synthesis, development, and operation of a real-world system that satisfies, in a near optimal manner, the full range of requirements for the system”.

As a profession: “SE is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem: operations, cost, schedule, performance, training and support, test, manufacturing and disposal. SE considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs”.

Systems engineering is interested in a cross-discipline integration, for instance if a requirement changes it is able not only to trace it and analyzed the impact in an specific department but also in the whole company system, by this way early design mistakes can be avoided. Knowing the capabilities of systems engineering, the aim of reviewing this discipline is to find a solution within it that allows modelling completely the electrical machine development process from conceptual design to validation of tests results from a prototype taking into account requirement management, costs analysis, development time and information management.

Systems engineering cope with a broad concerns of disciplines see Fig. 2-1a, no engineering discipline is independent in a company in some way all of them are connected, thus the problem of systems of systems arrives (see Fig. 2-1b) where a system is link to other systems making it even more complex, the more systems the more complex. Systems engineering has a holistic approach of these systems and is able to model the entire system, lately it is commonly applied in large industries like in automotive and aircraft manufacturers.

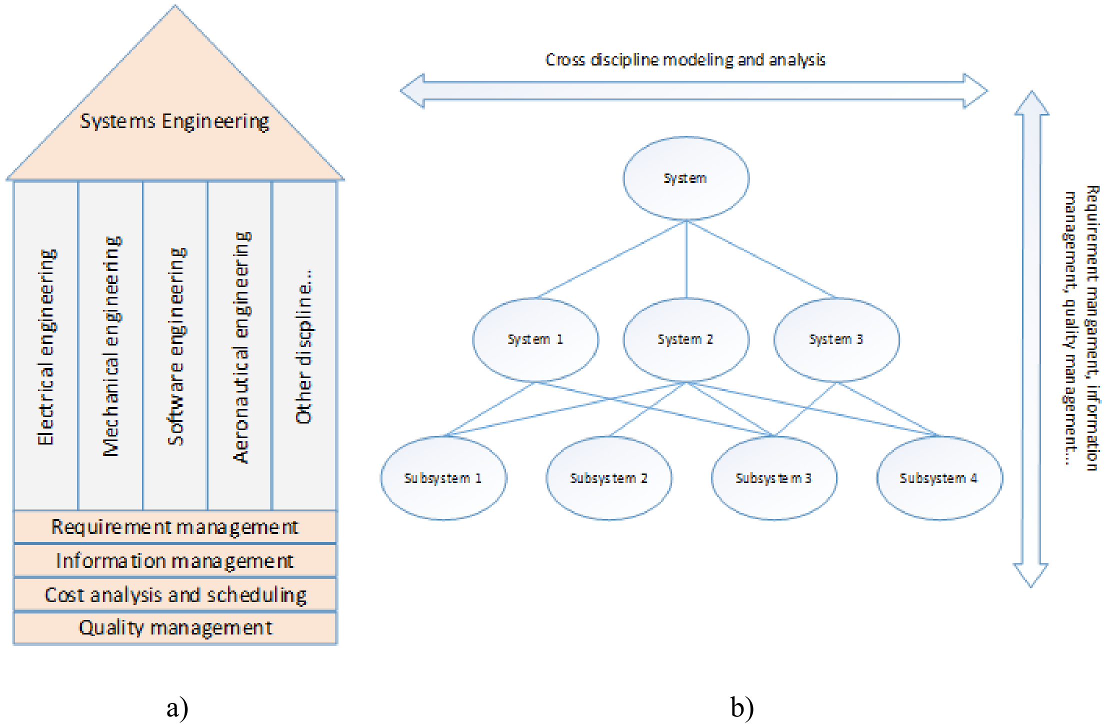


Fig. 2-1. a) Systems engineering cross-discipline perspective, b) Systems of systems complexity. Adapted from [65].

Several SE processes have been developed, such as the V-model from [66] or the SIMILAR process model [67] which gives a good overview of SE, see Fig. 2-2. A review of some other processes and standards are introduced in [68].

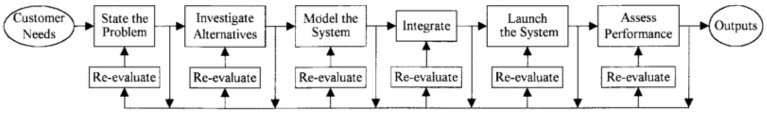


Fig. 2-2. SIMILAR process [67] p. 518

Systems engineering have been using the document-centric approach to model systems but lately the shift to model based systems engineering is commonly seen. A model is a representation of something, to represent something a language is needed either words, drawings, etc. The state of the art of these concepts, modelling languages and model based systems engineering are presented in the following subsection.

2.2.1 The System Modeling Language (SysML)

Modelling languages started for software development, now that used is extended to other areas. The Object Management Group (OMG) united efforts with INCOSE to develop a standardized extension of the Unified Modeling Language (UML) to be used as a modeling language for complex systems, this is SysML. SysML is relatively a new accepted standard, since 2006, OMG website [69] defines SysML as:

“The OMG systems Modeling Language is a general-purpose graphical modeling language for specifying, analyzing, designing, and verifying complex systems that may include hardware, software, information, personnel, procedures, and facilities. In particular, the language provides graphical representations with a semantic foundation for modeling system requirements, behavior, structure, and parametrics, which is used to integrate with other engineering analysis models. SysML represents a subset of UML 2 with extensions needed to satisfy the requirements of the UML™ for Systems Engineering RFP as indicated in Fig. 2-3. SysML leverages the OMG XML Metadata Interchange (XMI®) to exchange modeling data between tools, and is also intended to be compatible with the evolving ISO 10303-233 systems engineering data interchange standard”.

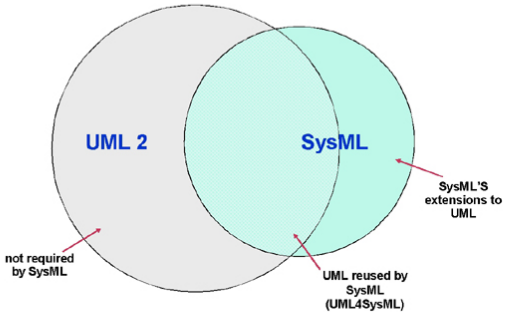
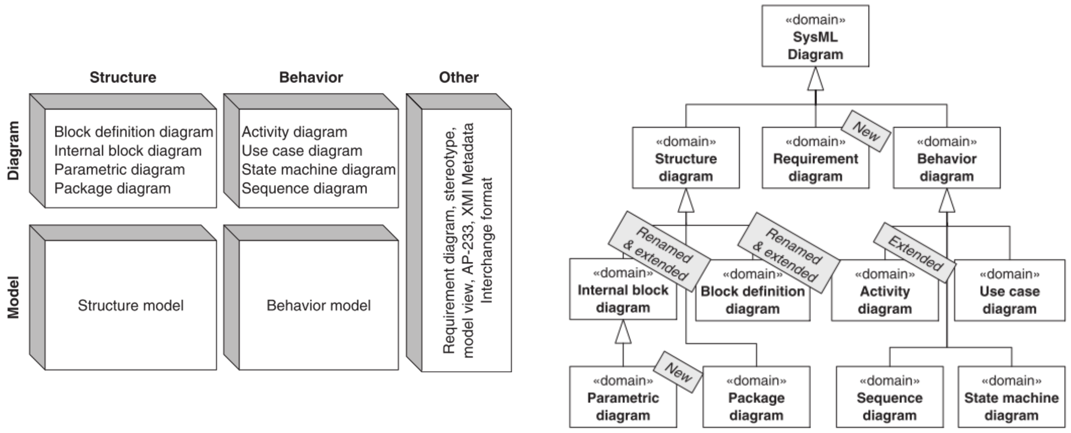


Fig. 2-3. SysML and UML space relation [69].

SysML consists of different diagrams based on behavior, structure and requirement, the structure of SysML as well as the diagrams are shown in Fig. 2-4. Other modeling languages are available the state of the art of these are UML/SysML and the Object Process Methodology language which was recently adopted in 2014 by ISO, in its standard ISO 19450 [70]. Dr. Dovi professor at the Israel Institute of Technology and MIT developed the language. The language is simple and easy to learn. However, due to the lack of software tools still available on market SysML is the chosen language for this thesis.



a) b)

Fig. 2-4. a) SysML structure. b) SysML diagrams. [71] p. 226, 227

Finally, it is important to state that one of the main contribution of SysML compared to UML is the introduction of the requirement as a model element allowing having traceability, satisfaction and derivations, facilitating early validation of design. To limit the content of this section, if the reader is not familiar with the language please check these relevant resources [71]–[73].

2.2.2 Model-based systems engineering

Model based systems engineering (MBSE) according to INCOSE definition is “the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later lifecycle phases”. [2]

MBSE is a subdiscipline of model-driven engineering (MDE), another subdisciplines of MDE are ontology engineering, model-driven software engineering and business process modeling, all specialized in systems engineering. SysML and UML has been the standard modeling languages. MBSE allows shifting from document-based systems engineering which made difficult to support the lifecycle phases due to the lack of evolution and integration with the system. Design information spread across many documents tends to inconsistencies. MBSE allows improving: information management, communication among the different disciplines involved in the system, quality supervision and knowledge capture by representation of it in standardized modeling language that allows reuse and reduce costs and cycle time to modifications in the design. According to [74] p.8, the advantages of MBSE compared to document-based systems engineering are the following.

Model requirements:

- *Technology advantage*: “ensure the requirements are an integral part of the model and all other parts of the model can be traced back to requirements”.
- *Business advantage:* “validate that you are building the right system”.

Model analysis and design:

- *Technology advantage:* “provide a precise architectural blueprint organization by the views that are meaningful to all systems stakeholders”.
- *Business advantage:* “Verify that you are building the system the right way”.

Model simulation:

- *Technology advantage:* “Automate system verification and validation”.
- *Business advantage:* **“**Reduce errors and costs early in the lifecycle”.

Model Code:

- *Technology advantage: “*Automate generation of production quality code”.

- *Business advantage:* **“**Accelerated time to market”.

Model Test:

- *Technology advantage: “*Automate testing”.
- *Business advantage:* **“**Ensure system implementation is correct and reliable”.

Nowadays several commercial tools can be found for the implementation of MBSE. Some examples of MBSE tools are:

1. NoMagic Cameo System Modeler
2. IBM Rational Rhapsody
3. Vitech Genesys
4. Enterprise architect (Sparx systems)
5. Scade (Ansys)
6. Artisan Studio (Atego)

In this thesis, based on the availability of the tool IBM Rational Rhapsody is selected. However, any tool could be selected regarding the purpose of this work.

2.3 A standardized macro-level framework for the development of electric machines

This section presents a standardized macro-level framework for the development of electric machines. The concept is based on the German guidelines VDI-2221 and VDI-2206. The goal is to represent the electric machine development process with a systematic approach in order to make it flexible to be adapted to any type of machine development project. As well as to allow the integration of state of the art software tools, to enhance the development process in industry.

2.3.1 A review of design methodologies with a systematic approach

Engineering design involves iterative decision-making processes where one or different engineering disciplines are applied to obtain a solution for a given problem. Many models can be found in literature about engineering design. These models can vary depending the discipline, engineers' personal choices and usually is a matter of the application and the know-how gathered from the engineering experience. Thus, a vast number of design processes flow diagrams are offered for all engineering disciplines [75]–[80]. This section highlights design methodologies that offer a systematic design approach allowing applicability in any discipline. The aim is to find a design methodology with a systematic approach that can be applied in electrical machine design and easily be adapted to any type of machine topology.

Design methodologies with systematic approach were collected from review sources [75], [76], [81]–[84], from which the common ones were separated and compared, see Table 2-1. Fundamentally, any of the selected methodologies could be applied, however, to make a better judgement some advantages and disadvantages were added regarding the scope of this thesis. The main drawback among all the selected design methodologies is no prototyping and testing phase is suggested with clear description. Some of them implicitly mention it but with not enough details of the process. In electrical machine industry in order to validate a new development, a prototype has to be built and tested.

The methodologies were analyzed based on replicability, adaptability, usability and effectiveness for electric machine applications. With replicability, what is wanted is to be easily duplicated, with adaptability that can be suited for the electric machine development process, with usability the intention is to check in some way how many industries and academics have applied it for any given design, and with effectiveness to see if those who applied it achieved their goals.

Table 2-1. Comparison of design methodologies with a systematic approach.

Methodology by	Description	Phases	Advantages	Disadvantages
Pahl and Beitz [75]	First time published in 1977 in the first German edition of "Konstruktionslehre" by Gerhard Pahl and Wolgang Beitz.	1. Task clarification, 2. conceptual design, 3. embodiment design 4. detail design	• Acceptance among industry and academics. • Descriptive, replicable and adaptable.	• Prototyping not included. • Test phase not included • No multidisciplinary design described.
Asimow [142]	Published in 1962 by Professor Morris Asimow at the University of California.	1. Feasibility study 2. Preliminary design 3. Detail design 4. Planning for production 5. Planning for distribution 6. Planning for consumption 7. Planning for retirement	• Descriptive and replicable.	• Some phases are out of the scope of this thesis
Cross [76]	First time published in 1984 by Professor Nigel Cross at The Open University in UK.	1. Objectives 2. Functions 3. Requirements 4. Characteristics 5. Alternatives 6. Evaluation 7. Improvement	• Descriptive • Acceptance among industry	• Few applications found in industry (derived from academia).
French [143]	First time published in 1985 by Professor Michael J. French at the University of Lancaster.	1. Analysis of problem 2. Conceptual design 3. Embodiment of schemes 4. Detailing	• Acceptance among industry and academics. • Descriptive, replicable and adaptable.	• Prototyping not included • Test phase not included • No Multidisciplinary design described
Hubka and Eder [144]	First published by Hubka in 1982, head of design science at ETH University.	1. Elaboration of assigned problem 2. Establish function structure 3. Laying out 4. Elaboration	• Acceptance among industry and academics. • Descriptive, replicable and adaptable	• Prototyping not included • Test phase not included • No multidisciplinary design described
VDI 2221 [77]	Published in 1993 by the Verein Deitscher Ingenieure (The Association of German Engineers).	1. Clarify and define the task 2. Determine functions and their structures 3. Search for solutions principles and their combinations 4. Divide into realizable modules 5. Develop layouts of key modules 6. Complete overall layout 7. Prepare production and operating instructions.	• Acceptance among industry and academics • It is a Standard • Descriptive, replicable and adaptable • Multidisciplinary design may be adapted from phase 3.	• Prototyping may be included but is not part of the guideline. • Test phase not included

After this review, the VDI-2221 methodology is selected. It should be noticed that for the thesis purposes, it is more important to use one methodology rather than select the best methodology. This guideline is a well-known standard among industry and broadly accepted by academics. Nonetheless, it does not specify multidiscipline designs. Electric machine design and manufacturing implicate different domains analysis such as mechanical, thermal and electromagnetics. To overcome this, a search was made for a methodology allowing multi-domain design and that it considers the prototyping and testing phases. Following the same line with VDI the VDI 2206 [85] was found. A guideline for mechatronic designs that has as complement VDI 2221 and can be well suited for electrical machines. Fig. 2-5 shows the VDI-2221 flow diagram and Fig. 2-6 the V-model.

A merge of the two models is proposed and described in the next subsection, several examples of merging the V-model with VDI 2221 or other models can be found in literature [86]–[88]. The purpose of merging the two models in this work is to:

1. have a design process model suited for electric machine with verification and validation of requirements along the whole process.
2. reduce iterations cycles of the V-model, instead, the intention is all iterations are made within VDI-2221.
3. implement multi-domain design.
4. allow the use of systems engineering for the whole development process.
5. have an attractive V-model that transmit most of the development perspective.

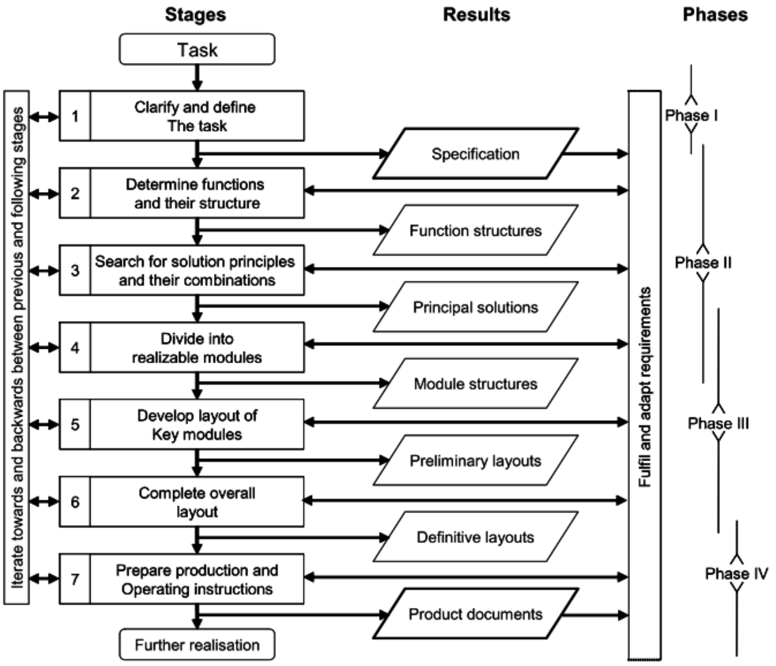
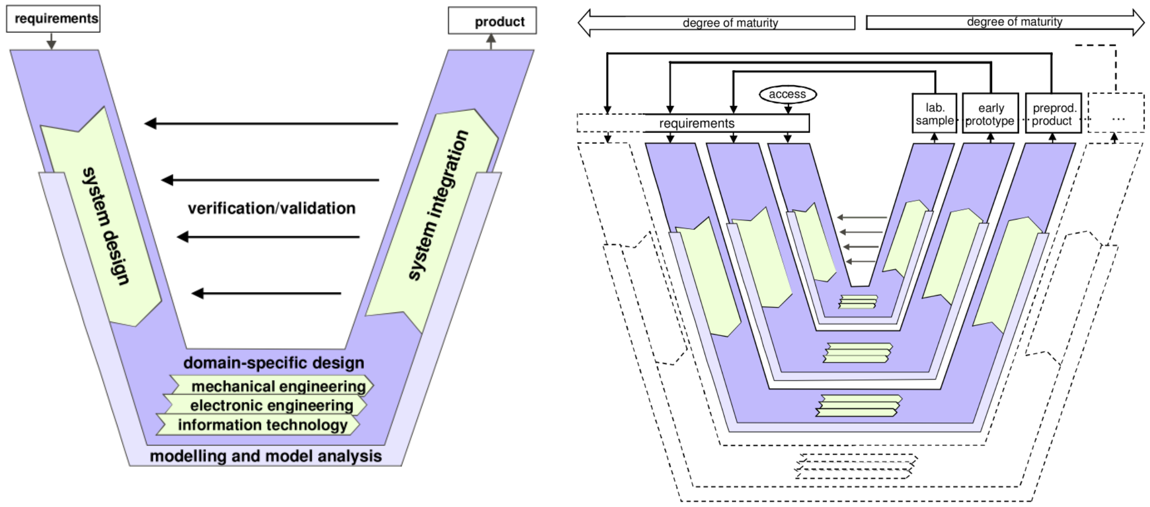


Fig. 2-5. VDI-2221 flow diagram from [89].



a) b)

Fig. 2-6. VDI-2206; a) Macro level V-model; b) V-cycles for product maturity. From [90].

2.3.2 The V2-model, merging VDI-2221 and VDI-2206 for the development of electric machines

Following both standards and the commonality in the design process of electric machines from literature described in Section 1.5.2, the proposed standardized macro-level framework for the development of electric machines is shown in Fig. 2-7. The author calls this model the V2-model because of the integration of the two VDI guidelines. The standardized macro-level framework attempts to embrace the development process providing a holistic view of the entire process and flexibility for adaptation in industry from both VDI standards. VDI-2206 supports the problem-solving cycle as micro-level and macro-level views. The macro-level view is represented by the V-model, on the other hand, the micro-level view is represented by the diagram shown in Fig. 3-25a. Both views are of great importance for this work, in particular for knowledge representation (Chapter 3). The aim is the micro-level cycle can be applied to solve any task within each step of the macro-level cycle.

The steps within the V-model in Fig. 2-7 were adjusted following VDI-2221. The iteration among stages and the fulfilment and adaptation of requirements is represented through the V-bar and red bi-directional arrows. Furthermore, model-based systems engineering is used for describing the processes without ambiguities.

The verification of requirements should take place during the whole process since their adaptation is also allowed during the whole process. The initial requirements entering the V-model are the initial requirements specification given by the client, from which the electric motor requirements should be derived for the design process. The outcome of the process is a validated machine for a specific application that depending on the initial objectives can become a product or just a step towards an end product. More details about this subject is explain later in this section. Now each stage is described.

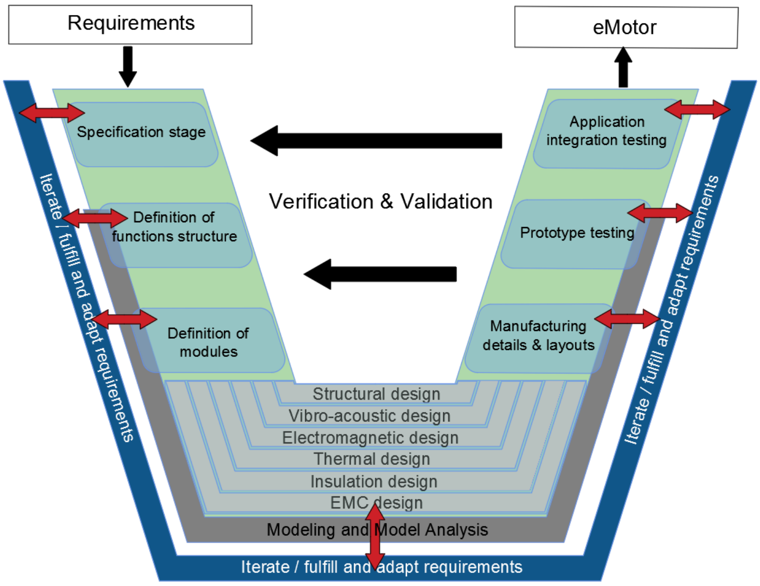


Fig. 2-7. V2-model, adaptation of VDI-2206/2221 for the electric machine development process.

2.3.2.1 *Specification stage*

The first step is the specification stage. In this stage the clarification and definition of tasks from the initial requirements specification from the client takes place. The process of identification of the design requirements depends on the application. Therefore, a study in-depth and assessment of the application is required in order to derive the correct design requirements of the machine. The correct execution of this stage is crucial to avoid unwanted redesigns in further stages.

For this reason, a requirement engineering framework is proposed in Section 2.4 describing the actors, processes, qualifications and tools to complete this phase successfully. Furthermore, this framework is a contribution to the lack of literature regarding this subject.

2.3.2.2 *Definition of functions structure*

The main function of the machine as an energy-transforming device is always the same. The focus should be put on variants of functions depending the application. Where depending the operating conditions on the application and the electric supply of the electric machine, it has to have different performances. In addition, there could be different operating modes in function of the protection signals and in faulty cases. Applications such as in aerospace industry requires fault-tolerance functions where the electric machines face to a faulty case (i.e. one phase open circuit) has to maintain to work in a certain level in degraded mode without having to be stopped. Normally, the client defines the different operating modes and conditions. The task for the designer will be to systematize all the information in a quantitative way using magnitudes of the electric machines and to delimit the different functions that defines a characteristic operating mode of the electric machine.

2.3.2.3 Definition of modules

Once the functions are set, the modules selection and arrangements that fulfils the functions are sought. This section takes into account availability of materials, manufacturing processes, modelling tools, expertise and others. Again, depending the initial objectives, re-use of assemblies or subassemblies can be implemented for faster design and manufacturing. For instance, re-use of end bells, bearings or stator and rotor lamination, and others. The degrees of freedom for the selection of modules are subject to the degrees of freedom set on the functions structure. In an electric machine, in order to properly define the modules, it is recommended to select the type of electric machine (induction machine, synchronous permanent magnet, etc.), the topology (radial, axial, etc.), and the type of refrigeration (natural, air-forced, liquid-forced, etc.). In some cases, more than one option could be interesting. In this case, more than one solution can be selected defining for each case the associated modules.

2.3.2.4 Domain-specific design

To allow cross-domain design and to cover all the machine features. This work proposes six domains: structural, vibro-acoustic, electromagnetic, thermal, insulation system, and electromagnetic compatibility (EMC). These six domains encapsulate the mechanical, electrical and control engineering disciplines making the modelling process simpler to view allowing concurrent engineering. Furthermore, even though each domain may follow its own methodology the integration of models is necessary to validate the motor solution.

The electromagnetic domain defines the magnetic and electrical circuit of the machine. Ensuring the electric machine performance characteristics are satisfied in operating conditions. An example of electromagnetic design methodology is shown in Fig. 1-10.

The thermal design involves the correct performance of the machine in the temperature range at which it will be operating. This will require estimating copper and iron losses from the beginning of the design to ensure that the machine is below the temperature limit. In addition, the design of the cooling system will be carried out if necessary. One of the most common reasons of failure in electric machines is due to the degradation of the insulation system, the life time of a machine is shortened if the insulation system is not properly defined. Among the main functions that the insulation system must fulfill is:

1. Restrain Eddy currents: the solution principle of this is to use insulation in magnetic sheets, as technical solutions we have inorganic, organic and mixed type coatings.

2. Resist electrical stress: prevents short circuits and resists operating voltages, different solutions are available depending on the type of winding, however, in general, there is ground insulation, copper wire insulation, phase insulation, wedges, winding ropes, winding and encapsulation tapes (drip impregnation and varnish immersion).

3. Resist thermal stress: prevent insulation wear due to high operating temperatures. The appropriate insulation material or class should be selected to withstand the highest operating temperature.

4. Resist mechanical stress: the insulation must withstand the vibrations to which the machine will be subjected and withstand the manipulation of the machine (assembly, transport, etc.)

The structural design takes into account fatigue analysis, torsional analysis, mechanical strain/stress analysis, buckling analysis, definition of manufacturing process for components, determination of geometric tolerances, fitness settings, selection of bearings and materials. More details can be found in [47].

Vibro-acoustic design focus on satisfying the levels of vibrations and noise. Noise sources in rotary electric machines can be classified as electromagnetic, mechanical and aerodynamic. This domain evaluates the levels of vibrations and noise caused by electromagnetic, structural and fluid dynamic phenomena. Therefore the modeling in principle will be dependent on the noise source and some cases will need multi-physic simulation (electromagnetic+structural+acoustic, fluid dynamics+acoustic, etc.). The reduction of the vibro-acoustic levels can be done modifying directly the origin of the force sources (e.g. modifying the electromagnetic design) or modifying the structural properties of the electric machine (e.g. modify to change the elasticity of some pieces to move the values of the natural frequencies).

Due to fast switching of IGBTs and high rate of change of voltage in power converters, there is a degradation of electromagnetic compatibility (EMC) affecting the reliability and safety in the operation of other electrical systems, especially those with electronic devices. It is also the cause of leakage currents due to parasitic capacitances in electric machines. Electromagnetic interference is classified into two types: differential mode (DM) and common mode (CM), the first one involves high frequency currents among phases and the latter among phases and earth (housing). Electromagnetic interference are key factors that must be considered since the beginning of the design process of the electric motor. The main problems encountered due to high voltage derivatives are:

1. Ground currents through parasitic capacitances inside the motor.

2. Conducted and / or radiated noise.

3. Shaft tension and bearing currents.

Regarding the electromagnetic compatibility design, it is important to be able to determine the impedance characterization of the interference propagation path, the electric machine is key since it is the main propagation element. Therefore, the model in the frequency range at which the motor will be running must be determined. In general there are two models, the low frequency model (10 kHz - 1 MHz) and the high frequency model (1 MHz - 10 MHz), for the first one the equivalent circuit is made through the determination of resistance and inductances and for the second it is not only the determination of resistance and inductances but also parasitic capacitances. At the same time, due to the high frequency parasitic currents that circulates inside the electric machine could damage some critical component (e.g. in medium voltage motors fed by PWM converters bearings could be damaged by induced high frequency currents that circulates inside the bearings).

Taking into account the integrated multi-physic study, Fig. 2-8 shows a high-level concept of the design process; within each activity, more actions take place depending the application. The process can be divided in three major phases. Phase one is the preliminary design, here the multiphysics models are designed following the specified requirements for each domain. The process assumes the requirements for each domain were already specified. The initial decision is to see if concurrent engineering is possible. If it is, then different domain models can be designed in parallel, typically this is possible when an electromagnetic design candidate is reused. If not, then the electromagnetic domain should provide an initial candidate. In all cases, the electromagnetic domain defines the main candidate and redistribute it to the other domains. No strict integration of the different models is required to check overall performance but only a functional model that satisfies the expected behaviour of the devised machine. It is important to remark that not all domain models may be required for every project. When all models are ready, the process outcome is a preliminary design.

Phase two involves optimization based on overall performance. The models are integrated and analyzed for satisfaction of requirements. The preliminary design enters an optimization iteration process until the design is considered valid. Verification and validation of the overall design performance is important in order to ensure each domain optimizes features of their model towards defined objectives. The level of optimization should be limited considering design times and costs. The outcome of this phase is a valid machine design ready for prototype manufacturing.

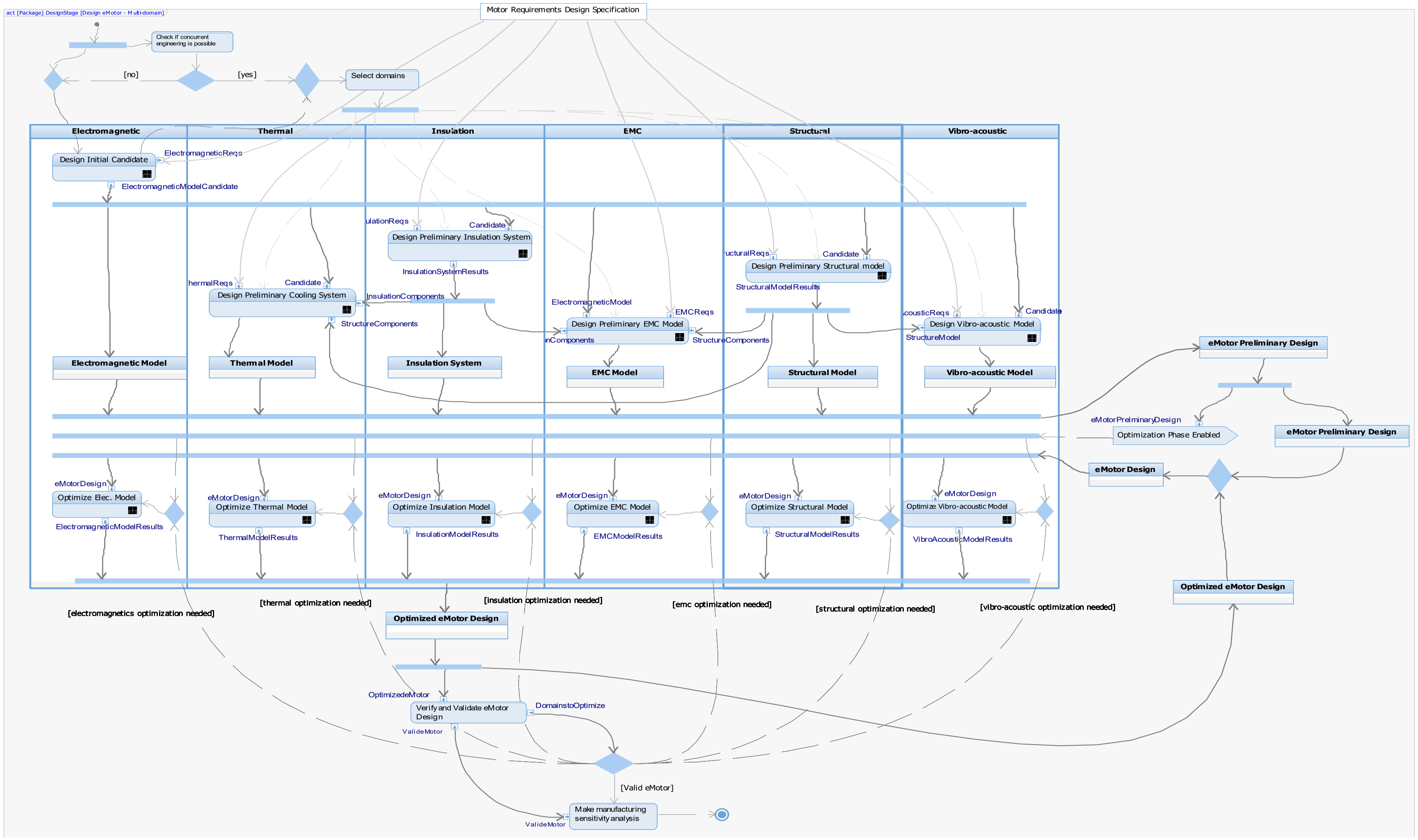


Fig. 2-8. High-level activity diagram for the domain-specific design stage.

Phase three includes the possibility of assessing a virtual sensitivity analysis considering manufacturing tolerances to ensure reliability of the process. Although the developed domain models in the previous phase are accurate, there still can be uncertainty due to manufacturing tolerances. Therefore, to obtain a robust design, parameters affected by tolerances should be examined. Gomez in [91] highlights and shows examples about this aspect and summarize in Fig. 2-9 the common manufacturing tolerances that has influence over output parameters that can affect the machine expected performance. The outcome of the process is a valid machine design solution subject to prototype testing for final validation.

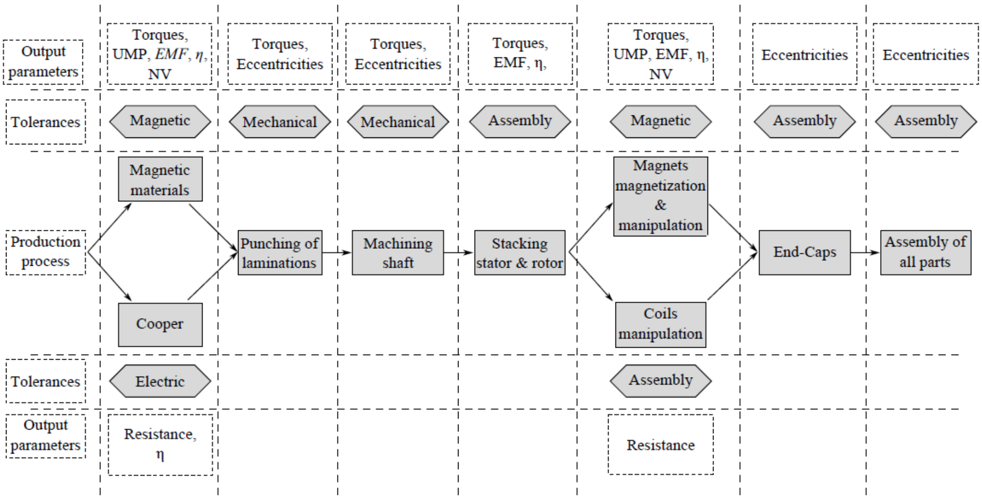


Fig. 2-9. Manufacturing tolerances that can affect the electric machine expected performance [91] p. 86.

2.3.2.5 Manufacturing details and layouts

In this stage, a complete manufacturing specification containing all the information for the realization of the electric machine prototype is developed. For instance, detail and assembly drawings, bill of materials, encapsulation, impregnation, lamination cuttings (stamped or machined), transport and operating instructions etc. The amount of information will depend if the manufacturing stage is executed by the own company or an external one. Usually, external manufacturers have define processes and they provide templates with parameters that must be fill out to execute the manufacturing order of the prototype. In addition, another important aspect to consider are in-process tests. In-process tests are tests performed to ensure the integrity of the machine parts before assembly.

2.3.2.6 Prototype testing

As part of the development process, building a prototype is compulsory in order to validate the performance and verify compliance with the design requirements, here is where testing start to come on the scene. For any application, it is essential to have a reliable and efficient operation of electric machines, testing is crucial to judge a correct performance. In serial or mass production of electric motors a series of tests are applied to each electric machine to ensure proper functioning. A variety of tests can be carried out with different purpose, for instance, to optimize the motor design, to detect faults, to improve reliability, etc. This work groups the tests in this stage into routine tests, type tests, special test, and research tests based on standards and present industry practice.

In summary, routine tests also known as series tests are applied on each motor with the intention to ensure product quality, confirming the basic acceptability and performance of a motor. Type testing or conformance testing are performed to check the design characteristics of an electric motor and verify its compliance with the specifications. Special tests are project specific, these are intended to verify a particular characteristic for the electric machine that is not included in the aforementioned tests. Finally, the research tests are perform to validate the design tools, manufacturing components or specific experimental design characteristics. These tests are common to be found in scientific literature. For instance, accelerated aging testing evaluates the reliability and sturdiness of motors with controllable faults to degrade materials, hence speeding up the normal aging, which yields to determine its lifecycle.

Finally, this stage must also have qualification of tests i.e. verification and validation that the test is executed correctly (e.g. mounting of the motor, bearings, noise, etc.), validation and verification that the results are with conformance with what is expected (e.g. analysis of curves obtained from resulting points).

Which tests to select depends on several factors such as cost, time, application and reliability. It should be noticed that for a personalized design in a short time to market competition, it will be not feasible to a apply a long time reliability tests. Then, new procedures are required in order to reduce the testing time for each new personalized design. Modular Designs and Virtual Validation Techniques are new trends to reduce the testing efforts for each new design.

2.3.2.7 Application integration testing

This stage considers tests in the application system. The electric machine is integrated into the application to ensure correct performance. Validation of performance and verification of requirements in the application is the last stage of the development process. If the stage is considered successful, we end up with a valid motor satisfying the purpose for which it was built. Nowadays, it is also very important to register and analyzed the performance and maintenance data of the electric machine along the whole life operation in the application. Until now, performance data were unknown because, it was not economically feasible to perform the measurements, register and collect an enormous amount of data. Nowadays, a new revolution has been introduced, associated to the Internet of Things (IoT), Web Computing and the Big data Analysis. These technologies allow the continuous monitoring of industrial equipment and to access remote field data easily. The knowledge extracted from field data will be very useful to improve the new designs allowing better optimization of the Key Performance Indicators (efficiency, power density, cost effectiveness, torque quality, mechanical robustness, RAMS indicators, vibro-acustic levels) of the electric machine to the real needs of the applications.

2.3.3 Macro-level V-cycles for the industrialization of eMachines

The purpose of this section is to include in the system the inherent difficulties towards an industrialization of an electric machine.

The whole process should include the following functionalities:

- **Innovation**: The designs should be opened to innovations inside and outside the company. It is necessary to increase the Design Space to the different options that open the new technologies, materials, etc.
- **Design for manufacturing**. The manufacturing cost of the designs should be very optimized to the client expectations.
- **Customization**. The designs should be easily adapted to improve the KPI of the electric machine to the application needs.
- **Short Time to Market Developments**. The time to place in the market a new development can be critical to reach a competitive position.

Typically, **the processes proposed in the scientific literature is based on a free design** where all the topologies, geometries and materials can be openly selected. These designs are not considering directly the cost of investments of the manufacturing equipment and tools (e.g. the cost of a new stamping die for steel laminations). At the same time the electric machines manufacturers tend to perform new designs trying to reuse as maximum as possible the equipment and tools available at the factory. These designs allows low investment costs, but the degrees of freedom of the design could be very narrow for a better performance optimization of the new product. Cost-effective **Product Customization** and **Short Time Developments** requires new paradigms in the design and manufacturing of the products. In the automotive industry, Modular Designs that can be used for different vehicles cover new designs and development investments. At the same time in this industry, the validation efforts are focused in the validation of the individual components. This way, once the component is validated for an industrial purpose with a high level demand of standardized requirements, this component is freely used to build a subsystem of the vehicle. This methodology reduces the efforts of the final integration test. This way, any component modification requires the full tests of this component and a light validation of the full subsystem. To perform the individualized validation of each component advanced model and experimental tools are used to take into account the global dynamic behavior of the subsystem or vehicle in the development of the component (Model-In-the-Loop (MIL) tools, Software-In-The-Loop (SIL), Hardware-In-the-Loop (HIL), advanced test-benches with dynamic emulated loads, etc.).

As was shown, the design of an electric machine is a multi-physic problem with many technical and scientific issues that should be considered at the same time for every specific design. Although there are very powerful multi-physic modelling tools, the full parametrization and modelling of full multi-domain problem requires huge efforts (including developing and testing laboratory prototypes). In order to reduce the design workload, it is recommended to use the previous know-how of the state of art about the design and manufacturing technologies.

In the case of an electric machine product, it is considered difficult to introduce a new technology in the market without testing and calibrating previously the new design solutions. In addition, it is important to iterate the new design solutions with the manufacturing process. For this issue, experimental testing and calibration tasks are needed. One option is to apply the three processes Innovation – Manufacturing definition – Final design in sequential tasks. In this thesis, this option is not suitable for a short time to market context. It is preferred to propose a division of the full process in three standardized steps:

1. *Concept Development Step*

In this step, the goal is to develop the innovative idea or new concept. For instance, a new topology is developed or a new material is introduced. It is proposed to follow a full design starting from requirements and ending with the testing of a prototype. The objective of this prototype is to validate experimentally the design hypothesis and the modelling tools and not to obtain a final product. The result is to obtain a Virtual Concept able to be modified or adapted to different applications needs.

2. *Component Development Step*

In this step, the goal is to develop the basic components applying the concept of Modular Design. Each basic component is associated with a specific manufacturing process and equipment. For example, in electric machine manufacturing a costly tool is the die to stamp the electrical steels. A typical cost of this “tooling die” could be around 200,000 € for an X-number of strokes resulting in the manufacture of Y-number of machines. Then it is highly recommended for a cost-effective fabrication to share the same die for different product ranges belonging to the machine developer. A typical cost-effective solution is to maintain the 2D magnetic shape and to extend the power level increasing the axial length. The process to industrialize each component could be laborious including hard iterations efforts with the specialized suppliers of the fabricated tools. The result of these tasks is to obtain the basic components and manufacturing tools. In order to validate these elements a complete design starting from requirements and ending with the testing of a prototype is proposed.

3. *Final Industrial Product Step*

In this step, the goal is to develop the final product that fulfils the specific requirements of the application of the client.

One competitive target is to be able to customize the electric machine to maximize the Key Performance Indicators that are more important for the client. At the same time, other competitive target is to reduce the time to put in the market the new product, trying to cope with a business necessity of the client. Then, in this thesis, it is proposed to customize the electric machine design using a design space of modular standardized components. This will make possible high degree of customization in short time to market developments.

Apart from the aforementioned three standardized steps, it should be noticed that all the development process is supported with advanced modelling and design tools that leverage the Virtual Design Development. In this sense, the developed Concepts and Components will not be only physical but they will also have a virtual definition. The virtual definition allows multiplying the number of possible final products using these Concepts and Components. Fig. 2-10. shows the scheme to expand the product solutions from Virtual Concepts and Components.

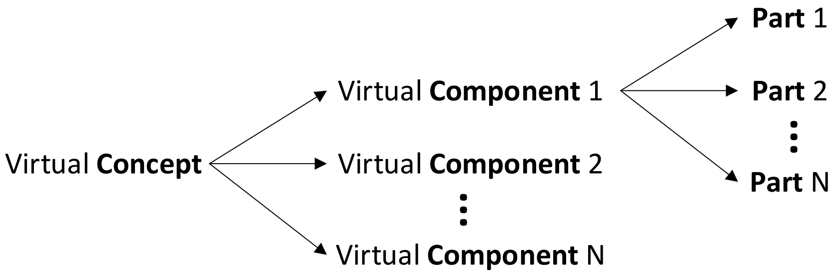


Fig. 2-10. Scheme to expand the product solution from Virtual Concept and Components.

In order to formalize the three standardized design steps, the VDI-2206 guideline is used to define the different degrees of maturity. Fig. 2-11. shows the proposed general scheme in this thesis. The three phases are concept V-cycle, component V-cycle and industrial V-cycle. All end up with a valid motor prototype, the difference arises in the degrees of freedom in design and manufacturing applied in the development process, and this is now explained.

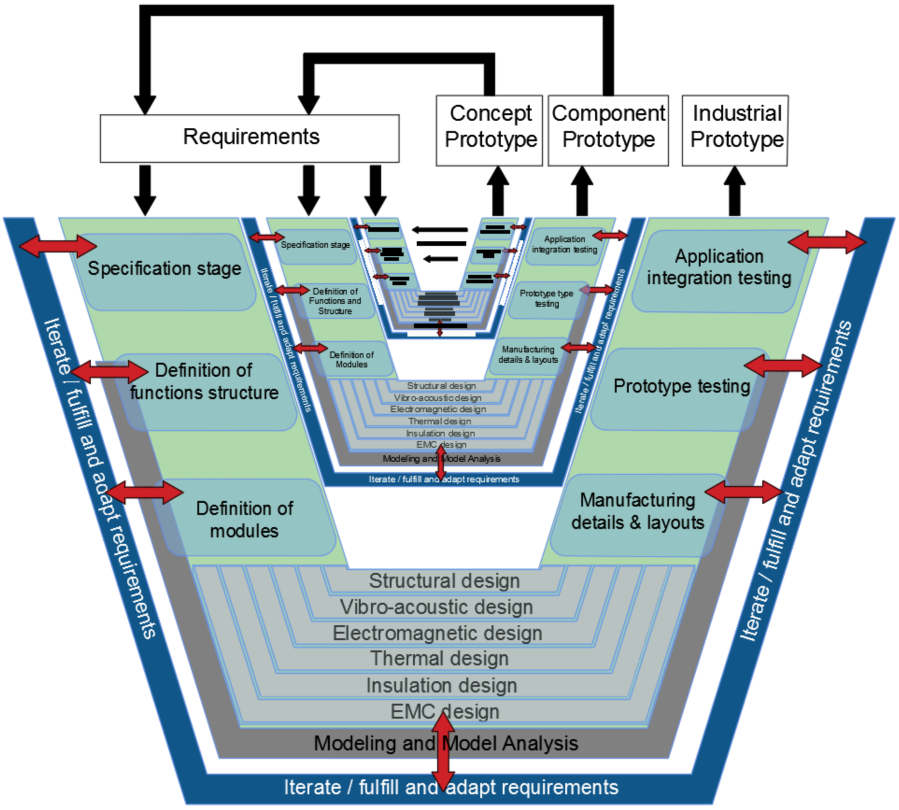


Fig. 2-11. Standardized macro-level framework for the development of electric machines.

2.3.3.1 Concept V-cycle

Transforming and idea into a physical machine concept. In this V-cycle, the degrees of freedom in design and manufacturing are unlimited. What this means is that the devised prototype is designed to guarantee technical viability. However, it may not guarantee economic viability as a final product. Typically, new concepts are built responding to customize solutions of applications. In this V-cycle, uncertainty regarding production can exist due to lack of industrialized parts. In other words, the different parts of the motor have no fixed and standard way of design and manufacturing, due to these are in the process of verification and validation.

Regarding the design process, a part can be design with a broad range of freedom such as geometry parametrization, material selection, and manufacturing process selection. A trade-off among the physical behavior should be analyzed in order to guarantee technical feasibility of the part. For instance in a rotor, embedded magnets increase the mechanical fixation robustness but if the application requires low speeds; magnets mounted on the surface can be a simpler solution of production. Another example is the stator and rotor lamination can be machined instead of stamped since creating a stamping die in this phase may be inappropriate since the concept is still in the validation process.

The basic characteristics of the Concept V-cycle are summed up.

Objective: To define and validate the virtual concepts of the electric machines. The full design is developed by joining the different virtual concepts.

Base Concept: It is a standardized conceptual idea having specific physical properties defined by quantitative parameters. Along the different possible parameters, it will be necessary to define the following parameters:

1. A full parametrized geometry with a wide range of degrees of freedom, but limited to the own definition of the virtual concept. For instance, for a surface permanent magnet concept the magnets can be inserted partially in the rotor at different heights as seen in Fig. 2-12.

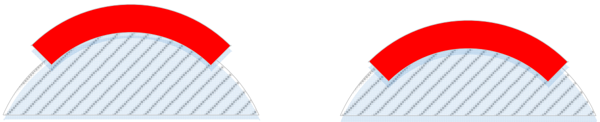


Fig. 2-12. Example of rotor lamination for a surface permanent magnet concept.

2. How to link children base concepts to parents base concepts. For instance, a concept of arrangements of magnets in the rotor can be obtained by the combination of different magnets position concepts to generate a new rotor concept.
3. Models that relate the concept parameters to a physical behavior. For example, depending the insertion of the magnets, on one side we have a different flux behavior and on the other side we have a better fixation of magnets making the design structural robust. Therefore, a model can include the electromagnetic behavior and another model can include the structural behavior. Fulfilling the entire physical behavior of the eMachine.
4. Define the different options of manufacturing that allows building and mounting the parts of the concept. For instance a general specification of manufacturing such as machined with laser or stamped either with slot-die or complete die.

One V-cycle is suggested to validate the complete concept design, several V-cycles can be executed for each new base concept.

The validation of the design process and manufacturing yields to the industrialization of parts. Thus, the next step is to work towards the reduction of costs and time while maintaining the quality of the product. This is done in the component V-cycle.

2.3.3.2 Component V-cycle

The goal is to transform an electric machine concept into a virtual industrialized part. The component V-cycle accounts experimentally validated ways of design and manufacturing. In order to create an electric machine prototype in this phase, previously, the definition of the design properties and manufacturing processes takes place. What this means is the range of materials are already selected, the stamping/cutting process is optimized as well as the definition of the lamination pattern, tolerances, grain orientation, and other manufacturing processes and testing of parts are defined. The construction of a prototype in this phase validates the already determined design and manufacturing components (e.g. stamping die, materials, impregnation, welding, coil winding process, etc.) for a given electric machine application.

By developing electric machines, another aspect of this V-cycle is the validation of created modular product platforms. Modular product platforms according to [92] consist of elements such as modules, assemblies or parts with defined standardized interfaces for several combinations to create variants efficiently. For instance you can use different combination of rotors stamping dies (i.e. topology) for the same stator stamping die to validate the performance of a motor with these variants and conclude if the combination can be considered valid for a specific application.

Different methods currently used in industry can be studied to design and build parts, these methods can be applied in this V-cycle for determination of processes. A summary of these methods is now made according to [75]. One of these methods is differential construction. Differential construction breaks down a component into smaller parts for simpler production or load-carrying capacity for large electrical machines. This method leverage the use of available semi-finished materials or standard parts, faster adaptation to existing factory layouts such as dimensions and weight and reduce production time because of faster adaptation to requirements compared to the concept V-cycle.

Another method is the integral construction. This method combines several parts into a single component for product optimization providing economic benefits due to faster production and savings in materials. Some examples are cast construction, and extrusions instead of welded construction and connected sections respectively.

In summary, some of the major differences in comparison to the concept are based on the degrees of freedom for example:

- In a component the manufacturing process and tools are already defined and experimentally validated.
- The lifetime of the component is already validated in all its geometry and properties range.
- The materials are restricted to commercially available materials that can be obtained easily from suppliers.
- Because of these previous features, normally the variation range of a component is less than a concept, e.g. in the case of a rotor sheet, a component can be an already

validated geometry for a given die design. In this case, the geometry will be fixed and it can only be adjusted to the material type and the lamination width within commercial options.

Using the manufacturing components e.g. stamping die, coil winding process, impregnation process, and magnets retention method, parts can be developed forming a pre-series prototype. This prototype allows validating in detail the technical viability of the components, although it would not be optimized in cost to specific specifications.

Finally, the basic characteristics of the component V-cycle are summed up.

Objective: Define and validate the base virtual components that will be used for a final industrial design.

Virtual Component: It is a practical application of the concept in which its degrees of freedom are restricted to industrial manufacturing taking into account the following.

1. The machinery and tools of the manufacturing process are defined and validated experimentally. These elements will define the tolerances and the range of geometric variation of the virtual component. For example, if a die is built for a rotor manufacturing, the rotor geometry is fixed and will be valid only for similar parts that share the same geometry with different type of materials.
2. The lifespan of the Virtual Component should be validated for the different variation range of geometry and properties.
3. The materials are restricted to commercially available products obtained from suppliers.

One V-cycle is suggested to validate a complete industrial design or several V-cycles for each virtual component (stator, rotor, etc.).

2.3.3.3 Industrial V-cycle

The goal of this V-cycle is to transform a virtual component into a real industrialized part. This V-cycle works with validated models and components for the development of electric machines. The objective of this V-cycle is to validate a prototype that later becomes an electric machine in serial production. A method closely related to this philosophy in the design process is the building block construction method [75] where validated parts are reused in order to produce product variants shortening the lead time. Within the optimized development process to specific specifications, the modules of the motor are selected making sure they satisfy the customer requirements. Each part of the motor has specified manufacturing plans, materials, processes, mounting, etc. Therefore, after validation of the prototype for the application the electric machine can be manufactured in series.

The basic characteristics of the Industrial V-cycle are summed up.

Objective: Define and validate the final detailed design of each product. Obtaining manufacturing layouts, declaration of norms, regulations and requirements fulfilled as well as performance standard data to share.

The V-cycle for the industrial product should be executed for each individual product reference, although several product references can share some design phases (same requirements, same mechanical design, etc.).

2.4 A requirement engineering framework for the development of electric machines

The specification stage is the basis for the efficient development of any product. This stage, if properly managed, avoids future failures in the development process [93]. The first step in designing or manufacturing an electric machine is to specify its requirements explicitly. Large manufacturers usually create guidelines or custom-forms with the specifications customers should take into account. In this way, they can supply a motor fitting their application (either from stock or tailor-made) [94], [95]. In [96], it is stated that a motor technical specification should define performance requirements, identify reliability indicators describing the environment in which it operates, and state maintenance conditions to help the user with its management plan. [97] presents some factors that affect decisions making when purchasing motors: product quality and reliability, price, service quality, brand reputation, energy efficiency, lead times, and policies. The weight of each factor varies according to the perspective of the end-users, for example, original equipment manufacturers (OEMs) or distributors.

Literature offers general requirement engineering frameworks [98] as well as specialized in different disciplines [99]–[101]. However, there is a lack of specialized frameworks for electric machine development. Usually, literature regarding electric machine starts with well-defined requirements, so the phase of the specification stage is not mentioned.

Moreover, large electric machine manufacturers have enough economic resources to develop their own specification process. Therefore, to support small-medium size electric machines design/manufacturing companies to cope with the specification stage, this work aims to contribute to the field of requirement engineering for electric machines by providing a new macro-process framework with a systematic approach. The framework is made from the designer/manufacturer's perspective and it emphasizes the derivation of requirements from a higher level to lower level requirements.

The experts from where this knowledge was captured have used and proven the approach herein described on several industrial projects. Furthermore, a use case example is presented to demonstrate the use of the framework.

2.4.1 Current requirement engineering approaches on electric motors

This section provides a comparison of requirement engineering approaches available on literature, addressing implicitly or explicitly the specification stage on electric machines. As mentioned earlier, there are some factors that affect the decision making on selecting electric machines being dependent on the perspective of customers or end-users. Therefore, the definition of requirements is application-dependent. For this reason, this work classifies the requirement engineering approaches as off-the-shelf and tailor-made solutions based on the two types of solutions that manufacturers can offer. The approaches are described from the developer's perspective.

2.4.1.1 Off-the-Shelf Solutions Approach

Standards, such as IEC in Europe and NEMA in the United States, define general characteristics and performance tests of electric motors to facilitate their interchangeability. In addition, regulations/policies/directives define mandatory requirements leading to minimum energy performance standards (MEPS). Large manufacturers have developed off-the-shelf motors satisfying these standards allowing for a simpler and convenient motor selection for customers. In this case, the specification stage is a process of sales-purchase transaction of equipment. Classic ways to handle the specification stage are either to make available the datasheets of motors in stock or to provide forms (e.g., spreadsheet) with requirements fields that the customers must fill to present motor solutions [94], [95], [102]. Due to the interest of customers in factors, such as efficiency, reliability, and quality of the product [97], some manufacturers offer support to find a solution, which best fit their application, from their catalog. Customers may demand factory acceptance tests or quality certificates of the motors. However, the interaction among the stakeholders is typically a sales-customer relationship.

2.4.1.2 Tailor-Made Solutions Approach

Applications demanding customized-solutions of electric machines are increasing, common examples are the automotive and aerospace industry. Automotive manufacturers are even improving production processes to include electric motors in their value chain [103]. For tailor-made solutions, design requirements are derived through a detailed analysis, where both, client and designer/manufacturer are involved during the development process or life cycle of the motor. Depending on the case, an efficient specification stage may involve different domain experts to obtain the requirements. In [104], the author acknowledges the need to consider the point of view of all stakeholders for an integrated design approach. In [105], the author states the objective of the specifications is to translate the function of the driven-load into the terms of electric drives, providing a classification of characteristics of the system with examples. An application example demanding special requirements on electric motors is presented in [106].

In conclusion, from the literature review, it can be stated that there is not a systematic approach that defines the iteration among the different stakeholders that are involved in the requirement development process. At the same time, there is a lack of a systematic organization among the different requirements. To have a successful design without spending time on redesign tasks, it is highly recommended that different technical experts check in detail the feasibility of all requirements that are associated with their engineering domain. The technical reviewing can become hard to do if the process is not well approached. Typically, an unwanted result is the waste of time due to several re-analysis of the same requirement. Regarding electric machine design, the absence of requirement engineering with a systematic approach in the technical literature is mainly due to the fact that the requirements are presented—as given—with all the details predefined. This level of detail is hard to achieve in tailor-made solutions and is recommended to have a collaborative approach between the integrator and developer to define accurate and realistic requirements. Therefore, this section presents a requirement engineering framework with a systematic approach describing a collaborative iteration among the stakeholders (eMachine integrator and eMachine developer) as a contribution to the lack of literature regarding this problem.

2.4.2 The framework

In this section, the requirements framework is presented through the Systems Modeling Language (SysML) diagrams. It is broken down into five subsections: the stakeholders, classification of requirements package, activities, qualification of requirements, and tools. This requirement engineering framework aims to provide a specialized guideline to electric machine developers. Users may adjust, add, or skip any aspect of the framework to suit their project.

2.4.2.1 Stakeholders

The framework identifies two stakeholders, the integrator and the machine developer (designer/manufacturer). The integrator is the actor or actors that are interested in integrating the developed electric machine into the application for which it is designed. The integrator is considered the expert in the particular application; in other words, the integrator has a complete knowledge in the application system, but usually no expertise regarding electric machine design. The electric machine developer is the actor or actors responsible for the design/manufacturing of the machine satisfying the integrators' requirements. The generalization of the latter actor is shown in Fig. 2-13. This generalization shows that the developer can be represented by several actors with expertise in the different fields that are involved in the electric machine development process. The framework is made from the electric machine developer's perspective.

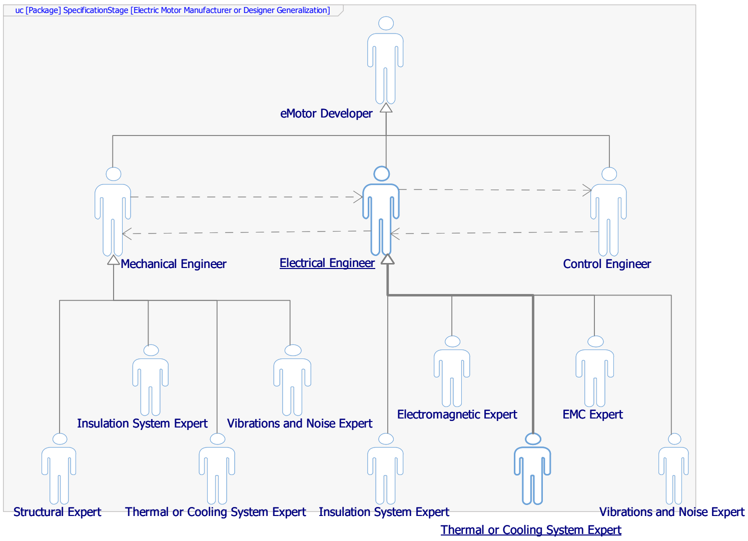


Fig. 2-13. Generalization of the electric machine developer.

The use case diagram shown in Fig. 2-14. illustrates both actors with their implications within the context of this framework. In summary, it can be seen that the integrator defines the machine requirements from the application point of view, which includes a clear description or explanation of the application system. On the other hand, the electric machine developer must derive the integrators' requirements into the different technical domains to develop the machine following an efficient process. This includes the process of interpreting each requirement; it may also include an engineering assessment of the application system, i.e., to model the application system to obtain its operating characteristics. Finally, both actors are involved in the approval, editions, or rejections of requirements during the entire development process.

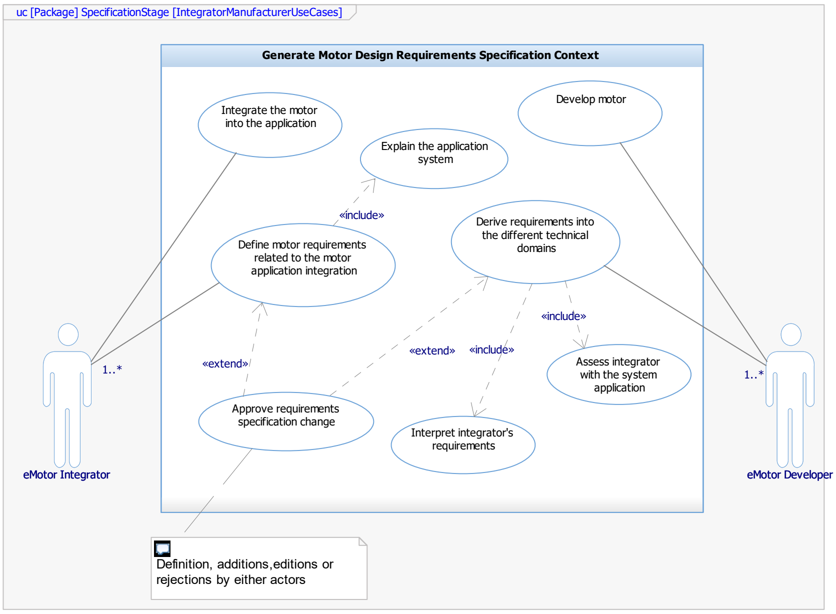


Fig. 2-14. Use case diagram of stakeholders in the requirements specification context.

2.4.2.2 Classification of requirements package

In order to facilitate the requirements analysis, the proposal is to divide the requirements into three general packages representing different levels, as it is shown in Fig. 2-15 and described below.

1. Application essential requirements from the integrator. The requirements contained in the initial specification are classified into different non-functional groups, such as operability, serviceability, reliability, comfort, special, and standards/regulations/policies requirements packages. The satisfaction of these requirements ensures the machine correct performance in the system application.
2. Technical derived requirements. These are derived or copied from the essential requirements into the different disciplines: mechanical, electrical, control, manufacturing engineering, also additional requirements and cross-domain objectives. The satisfaction of these requirements ensures the correct design and performance of the machine during the development process and the system application integration tests.
3. Domain-specific derived requirements. These are derived or copied from the technical derived requirements into the different domains that are involved in the machine development; herein emphasizing: structural, electromagnetic, vibro-acoustics, insulation, thermal, and electromagnetic compatibility (EMC). The satisfaction of these requirements ensures the correct design and performance of the motor during the development process.

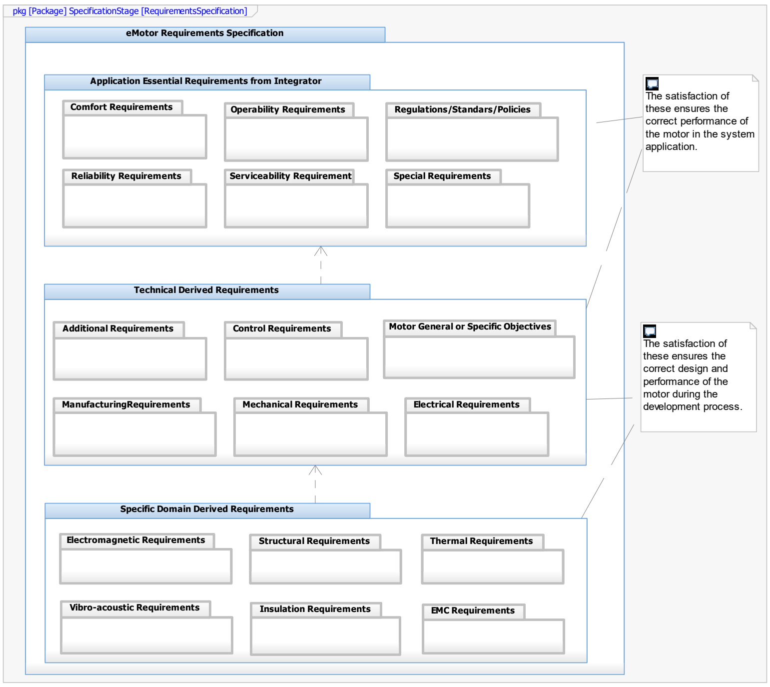


Fig. 2-15. Classification packages of requirements.

Table 2-2 shows examples of requirements that fit on each group. However, the developer (experts in their field) should decide this internal classification. The essential requirements may be found in compound form in the initial requirements specifications. If this is the case, to better define them, these ought to be decomposed into more requirements. Some may be implicit; nevertheless, if an analysis is carried out, these should be categorized as derived and put inside one of the technical derived disciplines group. Regarding the domain-specific requirements, these are the ones that each domain need to design its respective partition. Some requirements may only be copied from a higher-level group.

Table 2-2. Examples of requirements for each package.

Package	Requirements Examples
Application Essential Requirements Package	
Operability	Power sources, ratings, torque, speed, efficiency, service factor, duty cycle, environmental conditions, insulation, mounting, frame size.
Serviceability	Requirements related to ease of maintenance, ease of recycling, monitoring sensors (e.g., resistance temperature detectors (RTDs)).
Comfort	Noise and Vibrations, dynamic smoothness control, continuous operation smoothness.
Reliability	Life expectancy, permissible design or manufacturing tolerances, lead times, fault tolerances, tests.
Special	Transportation, training, mounting, assembly (e.g., on site), commissioning.
Regulations/standards/policies	Regulations, standards or policies that the motor must accomplish or follow.
Technical Derived Requirements Package	
Mechanical	Shaft torque, vibration and noise levels, types of bearings, rotor inertia, speed, shaft design, frame design, stresses, thermal constraints, cooling systems options, mounting constraints, enclosure type with the environment, insulation, size constraints, materials, sensors, related standards/regulations/policies, and others.
Electrical	Electromagnetic torque, winding arrangement, power factor, efficiency, voltage constraints, current constraints, size constraints, number of phases, insulation, thermal constraints, losses, torque density, electric motor type, materials, duty cycle or speed vs. torque curve, inductances, electromagnetic interferences, sensors, converter type, converter power ratings, related standards/regulations/policies, and others.
Control	Control strategy, switching frequency, converter type, sensors, phases, related standards/regulations/policies, and others.
Manufacturing	Machined lamination, stamped lamination, coil formation, encapsulation, impregnation, interference fit, rotor balancing, winding process, magnets mounting, magnets retention, magnetization, winding process, machined shaft, forged shaft, stack assembly, tolerances, related standards/regulations/policies, and others.
Machine General or Specific Objectives	Cost, materials, efficiency, lead time manufacturing, lead time design, motor type, life cycle, and others not included in other requirements classification.

2.4.2.3 Activities

The main activities of the proposed framework are shown in Fig. 2-16. The process starts when the integrator makes and handles the requirements specification of the application to the developer usually with explicit or implicit objectives. In this document, the regulations/standards/policies to be taken into account are also commonly included. After a quick inspection, if the developer finds the need for any information not available in order to start the requirement analysis, a request for information must be made. Otherwise, the next activity is to analyze the application requirements.

Generally, in this first stage, ambiguities are found, these must be avoided and well defined with the approval of both parts. For instance, “the motor must be highly efficient”. This statement is not quantifiable; therefore, a measurable quantity should be assigned either by a standard nomenclature (E2, E3, E4) or a restriction (>80%). The activity, analyze the application requirements, is composed of the different actions that are shown in Fig. 2-17. First, the developer should verify all the requirements and objectives from the integrator are within the application context in order to state the changes or to avoid additional work. Once this is done, the requirements must be classified into the different generalized non-functional groups that are described in Section 2.4.2.2.

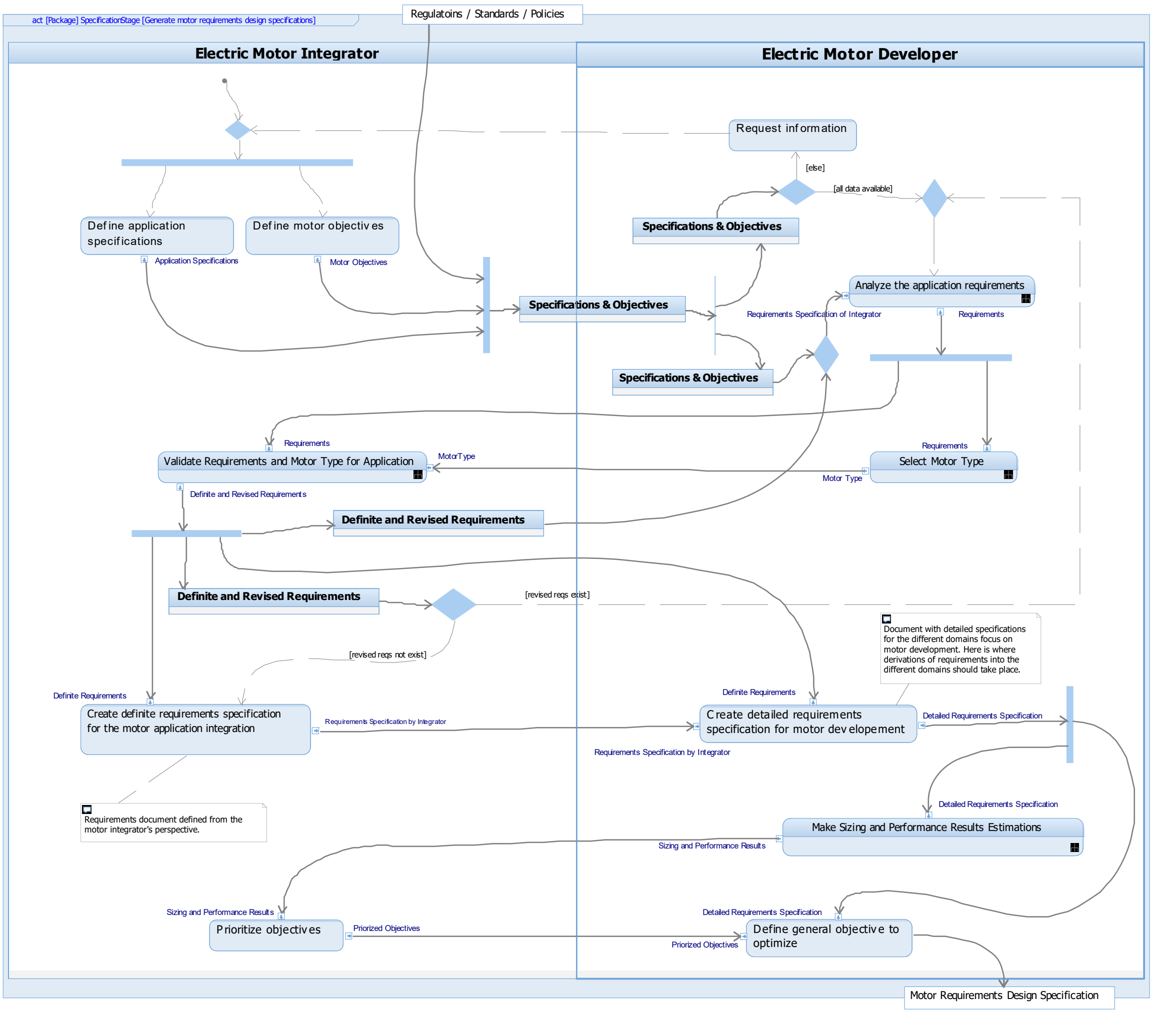


Fig. 2-16. Main activity diagram of the framework.

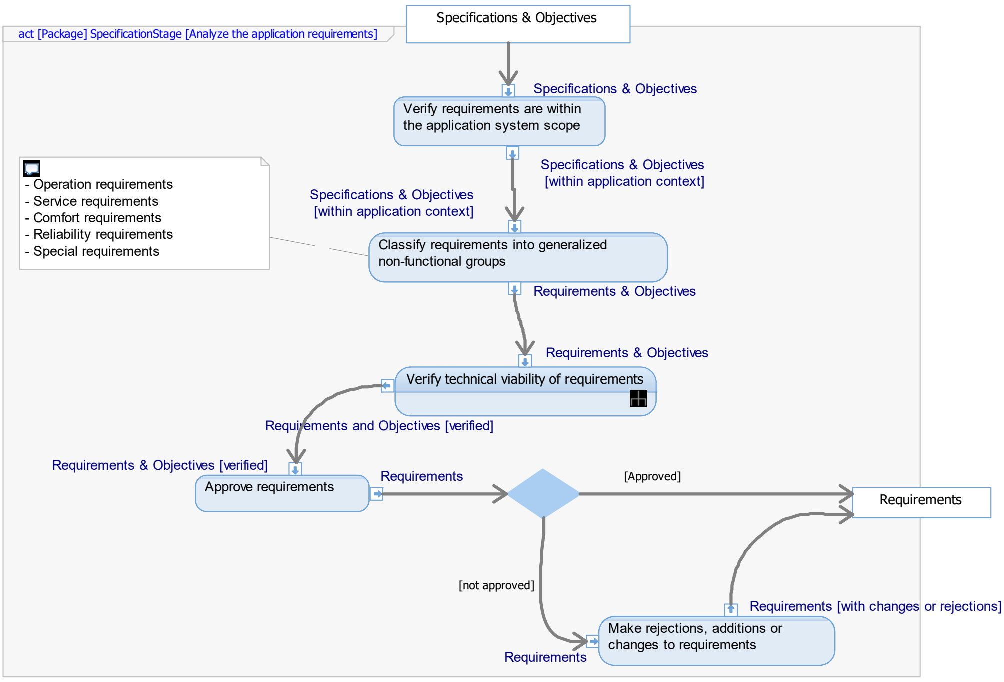


Fig. 2-17. Analyze the application requirements - activity.

The next activity is to verify the technical viability of the requirements that contain the actions shown in Fig. 2-18. To verify the technical viability of the requirements is necessary to derive requirements into the different disciplines involved in the project, these disciplines were described in Section 2.4.2.2. The analysis to derive requirements into the different disciplines is application dependent. This process may involve meetings, measurements on-site, testing, simulations, and all the activities that lead to obtaining the operating characteristics of the driven load. When the requirements are defined, their technical viability and cost feasibility must be verified. Once this is done, the result is a set of revised and verified requirements. Resuming the activities of Fig. 2-17, the requirement identification considering its approval, rejection, or modification takes place. The requirements must have an attribute that states one of the aforementioned options.

The next step in the main activity diagram (Fig. 2-16) is to handle the requirements that are of interest to the integrator to approve, reject, or modify them. In addition, the motor type must be selected if it is not explicitly defined in the initial document. This is done by comparing costs and technical criteria among the different motor technologies considering the one that best fits the application system. Examples can be found in [107], [108]. The integrator must validate the requirements and motor type. This activity is illustrated in Fig. 2-19. Definite or revised requirements are the output of this activity. Revised requirements must be transferred to the developer to be approved, rejected, or changed. This is a cycle that must end when both parts agree with the requirements (definite requirements) and when both parts consider the project is well-defined in order to start the design phase.

Once the definite requirements are available, it is good practice for the integrator to redo the requirements specification from his perspective. The next step is to create a detailed requirements specification for the motor development. It is here where the derivation of requirements to the specific domains that were mentioned in Section 2.4.2.2 take place. This derivation of requirements is done from the different disciplines to each specific domain. For instance, in the case that the design needs to have low levels of vibrations and noise (mechanical requirements), a requirement that limits the torque ripple can be derived to the electromagnetic domain. The integrator may specify the vibration or noise level but not a torque ripple constraint (if he is not aware of the term), so this job must be carried out by the developer.

The next activity (make sizing and performance results estimations) involves checking the possibility to make an early virtual prototype with previous designs or with analytical models to give the integrator an idea of what is possible to be achieved. What this step tries to accomplish is the validation in a preliminary way (not definite) of the space constraints, costs, materials, or any other parameter of interest. However, this is not always possible more so when the project is innovative. If it can be done, the integrator can develop new requirements or prioritize objectives. If objectives are prioritized, then the developer can better define general objectives to be optimized. At the end of the process, the developer has a motor requirements design specification that allows him to start the design phase. Obviously, the requirements, during the development process may be subjected to modifications or new requirements may appear. Therefore, the appropriate step must be retaken.

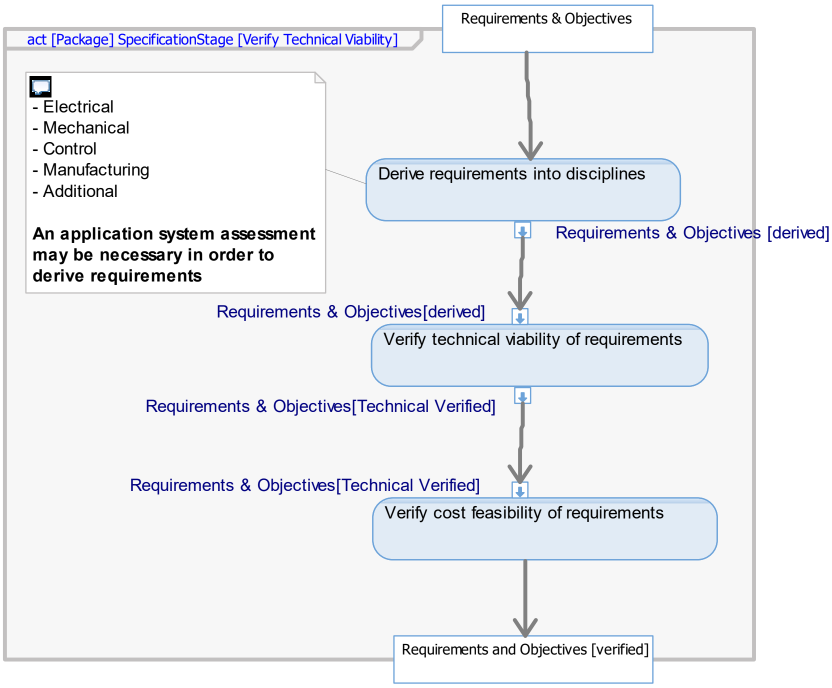


Fig. 2-18. Verify technical viability - activity.

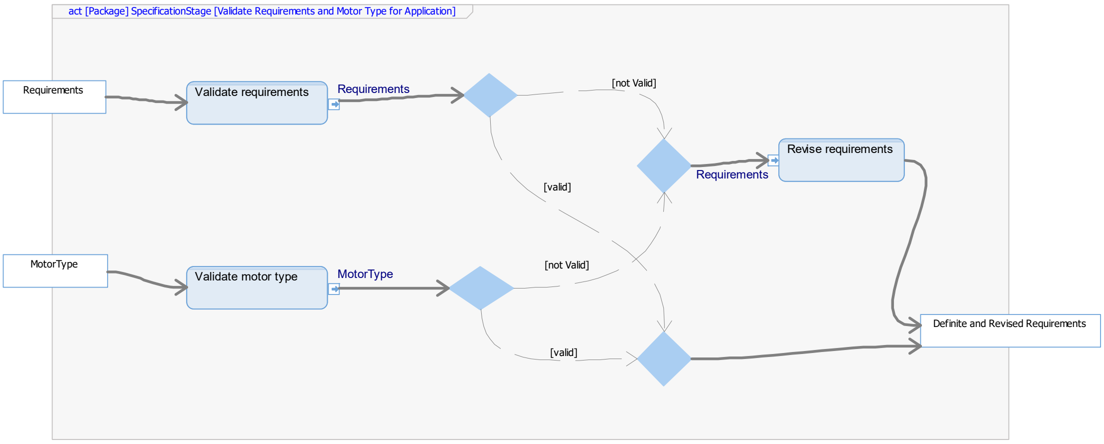


Fig. 2-19. Validate requirements and motor type - activity.

2.4.2.4 *Qualification of requirements*

The term qualification refers to the testing activity that ensures the solution has its conformance to requirements [98]. During the specification stage, tests that verify the satisfaction of the requirements and validate the design and manufacturing stages should also be defined. The tests are generalized into five different types:

1. Virtual prototype performance testing: These tests are done through simulations to the electric machine by the different domains. It should be carried out in the conceptual design phase.
2. Virtual system application performance testing: These tests integrate the electric machine into the application, but they are also carried out through simulations during the conceptual design phase.
3. Prototype performance testing: These tests are carried out on the manufactured prototype in a laboratory. These tests include type testing, routine testing, or any research testing.
4. Manufacturing tolerance testing: These tests are done to components when interested in serial or mass production.
5. System application performance testing: These tests integrate the electric machine into the application and ensure the correct performance of the machine with the load system. Previously, factory acceptance tests may be included only if the manufactured machine is considered to be the final solution. Fig. 2-20 illustrates the relationship among these generalized tests and the requirements packages. The realization of these tests should verify the satisfaction of the requirements within each package.

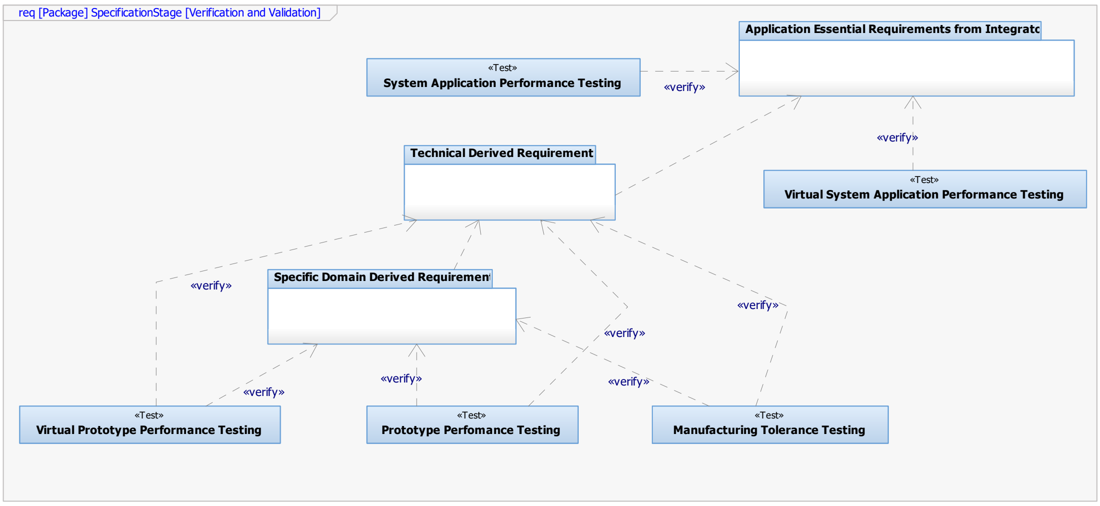


Fig. 2-20. Generalized tests and verification of requirements.

2.4.2.5 Tools

To carry out the framework, requirement management tools are recommended and, if available, model-based systems engineering (MBSE) tools. Regarding requirement management, several open-source and commercial tools are available, such as ReqIF Studio and Rational Doors, respectively. Available MBSE tools are also offered as open-source, like Papyrus, and commercial ones, like Rational Rhapsody, Enterprise Architect, Magic-draw, and others. The tools allow for the creation of matrices and tables that facilitate the view of relationships and the traceability of requirements.

2.4.3 Use case example: deriving electromagnetic requirements for an elevator application.

A use case example is presented regarding the request for a tailor-made electric motor for an elevator system application. The tools used are IBM Rational Doors 9.5 and IBM Rational Rhapsody 8.3.1. The mechanical system of the elevator consists of a 2:1 roping ratio with one cabin, one counterweight, a set of cables, and two pulleys. In this example, we focus the framework only on electromagnetic requirements derivation from given requirements handled by the integrator for the design of a surface permanent magnet synchronous motor. Herein, all requirements identification (ID) are described with the prefix elevSysPM, which belongs to the project name followed by the package owing the requirement, as shown in Table 2-3 and finally a unique number e.g. elevSysPM-InRS-01.

Table 2-3. Requirement package IDs.

Requirement Package ID	Package Name
InRS	Integrator's requirements
MechRS	Mechanical engineering requirements
ElecRS	Electrical engineering requirements
ControlRS	Control engineering requirements
ManufRS	Manufacturing engineering requirements
ObjRS	Set objectives during the development process
ElecmgnRS	Electromagnetic requirements

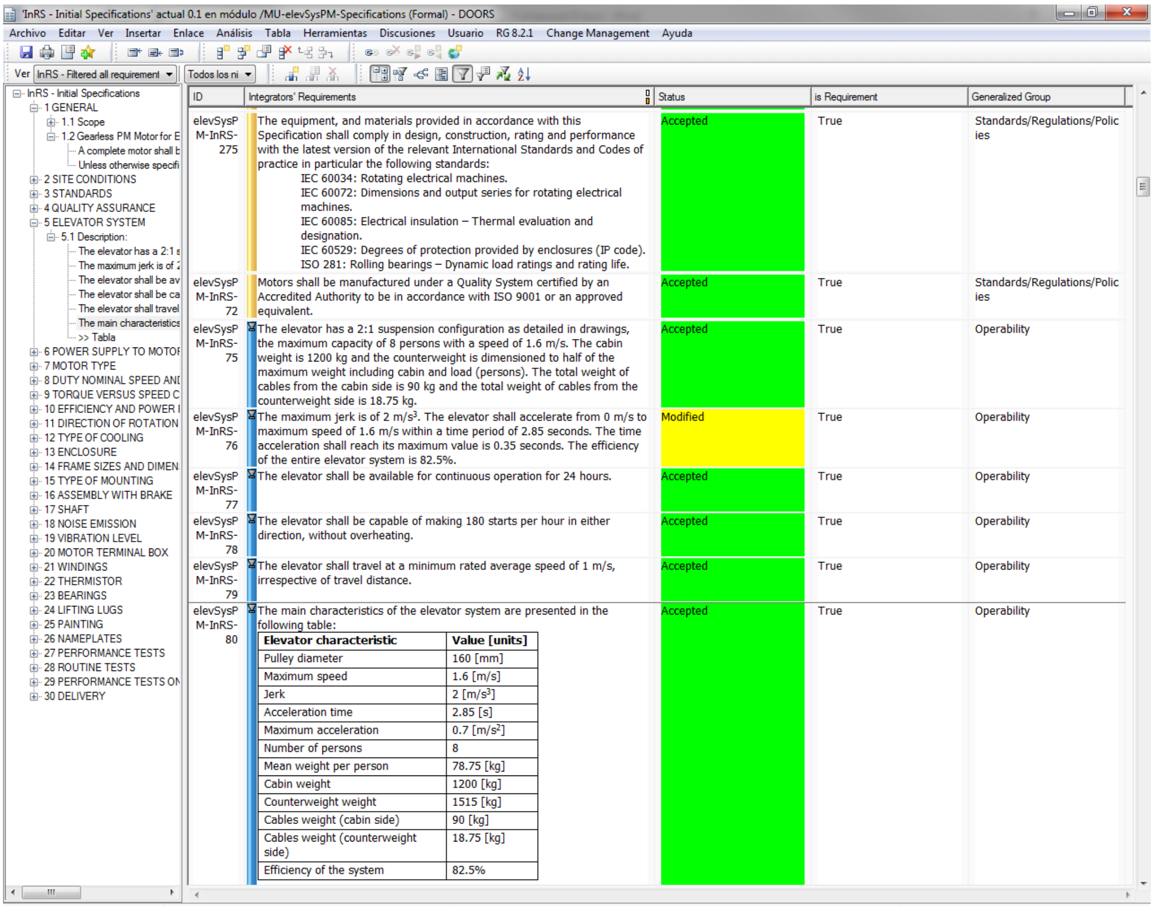
2.4.3.1 Requirements within the elevator system application scope

The initial requirements specification was provided in a word document. After its acceptance, this is imported to Rational Doors and a classification is made separating the non-requirements from requirements. Fig. 2-21 shows a screenshot of Doors where some requirements can be seen with their attributes. The same figure shows a classification of the requirements into the six non-functional groups: operability, reliability, comfort, serviceability, special, and standards/regulations/policies. The status attribute shown comes posterior to the verification of the technical viability of the requirements; nevertheless, it is shown as an example.

2.4.3.2 Deriving requirements into disciplines

The requirements are imported from Doors to Rhapsody to illustrate the process of derivation. In order to derive requirements into the different disciplines for this customized motor design, first, the operation characteristics curves (speed, torque, acceleration, jerk, and position) of the system must be obtained. A system assessment is carried out due to the initial specification not presenting these curves. The initial specification does present the important parameters to derive these curves. Fig. 2-22 shows the curves that were processed through parametric diagrams using the equations of Fig. 2-23. The equations that model the behavior of the system were obtained and agreed among the stakeholders and stored as constraint properties.

The operating characteristics curves were obtained for a trip distance of 7.2 m, i.e., two floors of 3.6 m in height. The trip ends in 8 s, and at that distance, the brake is activated releasing the motor from doing work. The minimum time that is required by the elevator system to stop is 12 s, time for doors to open, people to get in/off, and doors to close. Therefore, the worst case scenario thermally speaking is to study a cycle of 20 s. All the aforementioned information is captured in the diagram to illustrate the derivation of mechanical requirements.



ID	Integrators' Requirements	Status	is Requirement	Generalized Group
elevSysPM-InRS-275	The equipment, and materials provided in accordance with this Specification shall comply in design, construction, rating and performance with the latest version of the relevant International Standards and Codes of practice in particular the following standards: IEC 60034: Rotating electrical machines. IEC 60072: Dimensions and output series for rotating electrical machines. IEC 60085: Electrical insulation – Thermal evaluation and designation. IEC 60529: Degrees of protection provided by enclosures (IP code). ISO 281: Rolling bearings – Dynamic load ratings and rating life.	Accepted	True	Standards/Regulations/Policies
elevSysPM-InRS-72	Motors shall be manufactured under a Quality System certified by an Accredited Authority to be in accordance with ISO 9001 or an approved equivalent.	Accepted	True	Standards/Regulations/Policies
elevSysPM-InRS-75	The elevator has a 2:1 suspension configuration as detailed in drawings, the maximum capacity of 8 persons with a speed of 1.6 m/s. The cabin weight is 1200 kg and the counterweight is dimensioned to half of the maximum weight including cabin and load (persons). The total weight of cables from the cabin side is 90 kg and the total weight of cables from the counterweight side is 18.75 kg.	Accepted	True	Operability
elevSysPM-InRS-76	The maximum jerk is of 2 m/s^3. The elevator shall accelerate from 0 m/s to maximum speed of 1.6 m/s within a time period of 2.85 seconds. The time acceleration shall reach its maximum value is 0.35 seconds. The efficiency of the entire elevator system is 82.5%.	Modified	True	Operability
elevSysPM-InRS-77	The elevator shall be available for continuous operation for 24 hours.	Accepted	True	Operability
elevSysPM-InRS-78	The elevator shall be capable of making 180 starts per hour in either direction, without overheating.	Accepted	True	Operability
elevSysPM-InRS-79	The elevator shall travel at a minimum rated average speed of 1 m/s, irrespective of travel distance.	Accepted	True	Operability
elevSysPM-InRS-80	The main characteristics of the elevator system are presented in the following table:	Accepted	True	Operability

Elevator characteristic	**Value [units]**
Pulley diameter	160 [mm]
Maximum speed	1.6 [m/s]
Jerk	2 [m/s^3]
Acceleration time	2.85 [s]
Maximum acceleration	0.7 [m/s^2]
Number of persons	8
Mean weight per person	78.75 [kg]
Cabin weight	1200 [kg]
Counterweight weight	1515 [kg]
Cables weight (cabin side)	90 [kg]
Cables weight (counterweight side)	18.75 [kg]
Efficiency of the system	82.5%

Fig. 2-21. Initial requirements classification in Rational Doors.

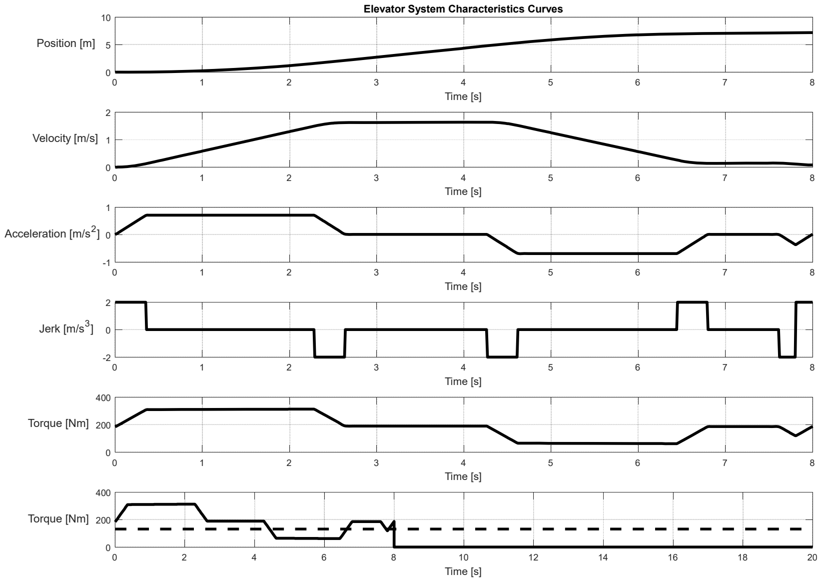


Fig. 2-22. Operating characteristics curves of the elevator.

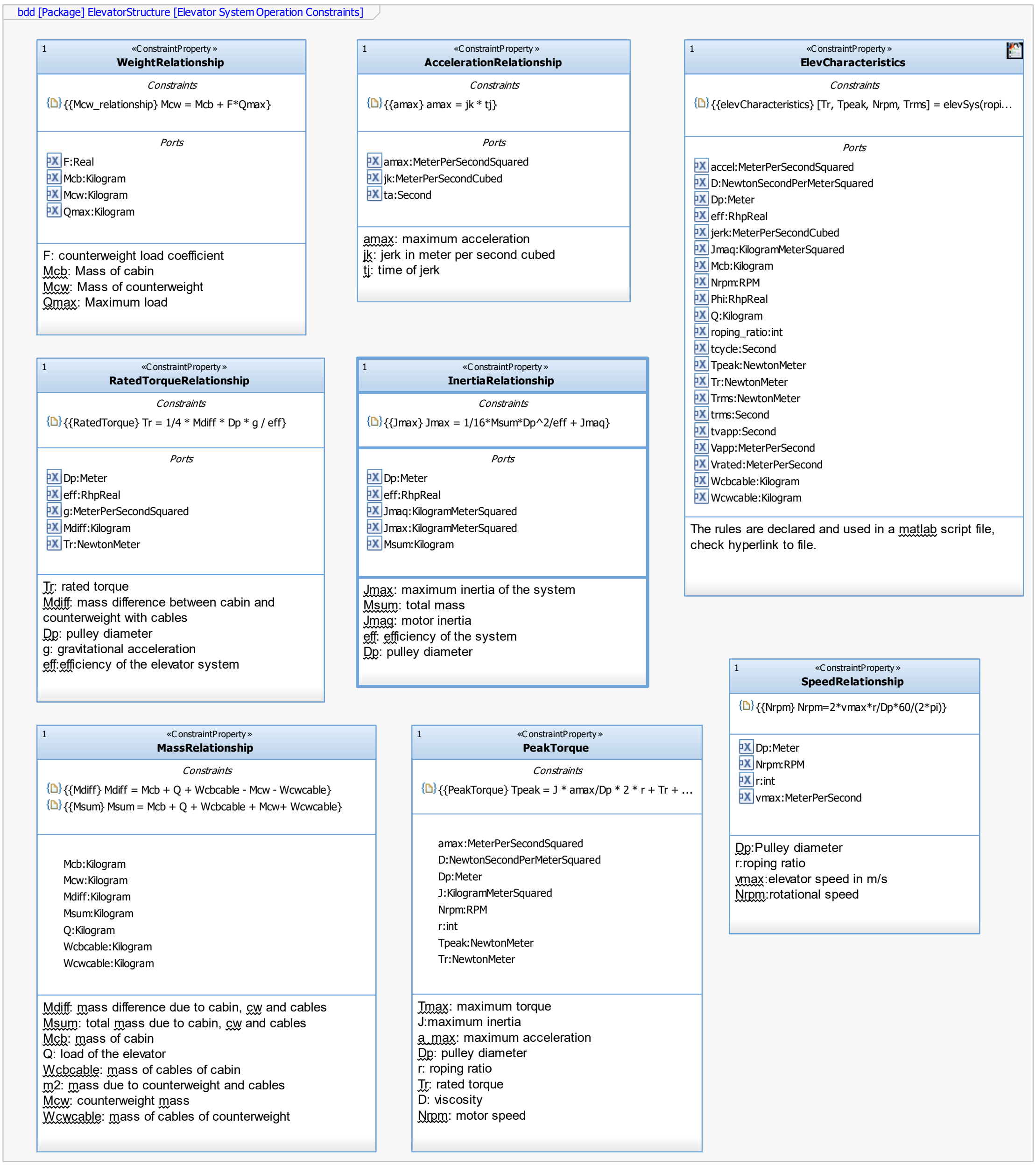
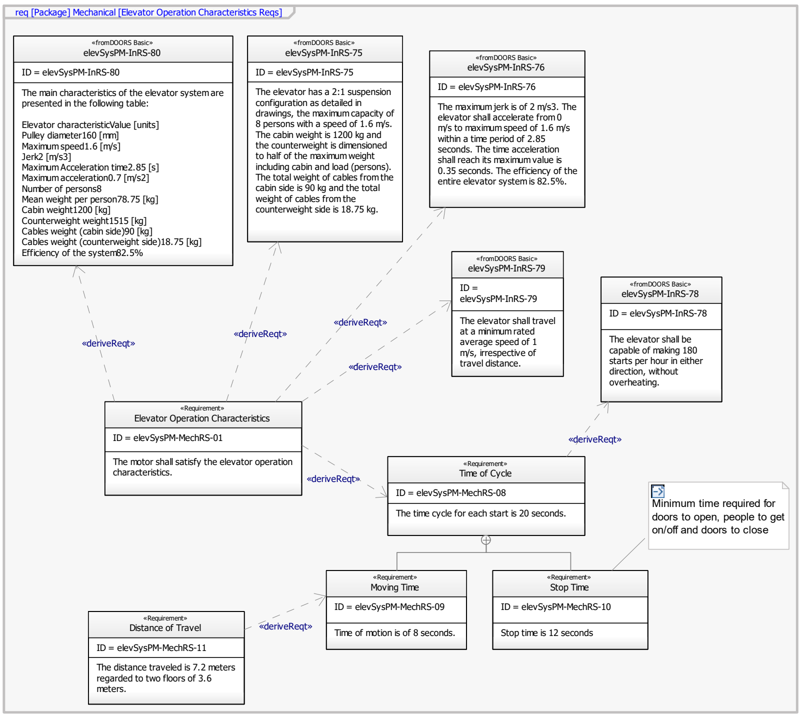


Fig. 2-23. Equations modeling the elevator system operating characteristics.

The requirement diagram in Fig. 2-24 shows the relationships of this elevator operation characteristics requirement derived as a mechanical requirement from the initial requirements. Fig. 2-25 shows the parametric diagrams that refine this requirement. Although not being shown in the figure, for each characteristic curve a mechanical requirement is also created, i.e., this requirement contains five mechanical requirements one for each curve. These are used later to derive the electromagnetic requirements.



ID = elevSysPM-InRS-80

The main characteristics of the elevator system are presented in the following table:

Elevator characteristicValue [units]
Pulley diameter160 [mm]
Maximum speed1.6 [m/s]
Jerk2 [m/s3]
Maximum Acceleration time2.85 [s]
Maximum acceleration0.7 [m/s2]
Number of persons8
Mean weight per person78.75 [kg]
Cabin weight1200 [kg]
Counterweight weight1515 [kg]
Cables weight (cabin side)90 [kg]
Cables weight (counterweight side)18.75 [kg]
Efficiency of the system82.5%

ID = elevSysPM-InRS-75

The elevator has a 2:1 suspension configuration as detailed in drawings, the maximum capacity of 8 persons with a speed of 1.6 m/s. The cabin weight is 1200 kg and the counterweight is dimensioned to half of the maximum weight including cabin and load (persons). The total weight of cables from the cabin side is 90 kg and the total weight of cables from the counterweight side is 18.75 kg.

ID = elevSysPM-InRS-76

The maximum jerk is of 2 m/s3. The elevator shall accelerate from 0 m/s to maximum speed of 1.6 m/s within a time period of 2.85 seconds. The time acceleration shall reach its maximum value is 0.35 seconds. The efficiency of the entire elevator system is 82.5%.

ID = elevSysPM-InRS-79

The elevator shall travel at a minimum rated average speed of 1 m/s, irrespective of travel distance.

ID = elevSysPM-InRS-78

The elevator shall be capable of making 180 starts per hour in either direction, without overheating.

ID = elevSysPM-MechRS-01

The motor shall satisfy the elevator operation characteristics.

ID = elevSysPM-MechRS-08

The time cycle for each start is 20 seconds.

Minimum time required for doors to open, people to get on/off and doors to close

ID = elevSysPM-MechRS-09

Time of motion is of 8 seconds.

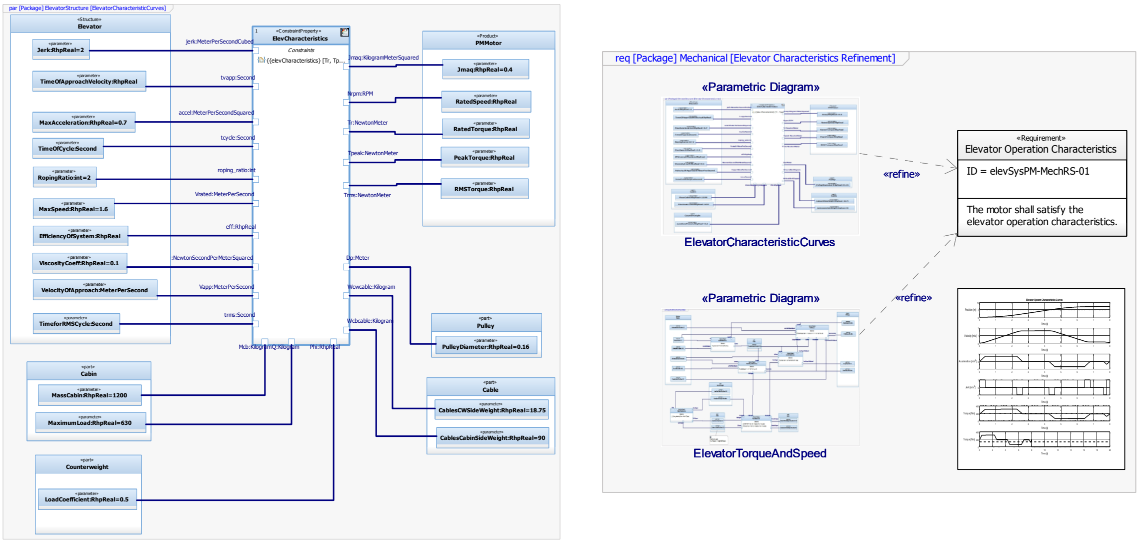
ID = elevSysPM-MechRS-10

Stop time is 12 seconds

ID = elevSysPM-MechRS-11

The distance traveled is 7.2 meters regarded to two floors of 3.6 meters.

Fig. 2-24. Mechanical requirements derivation.



a) b)

Fig. 2-25. Elevator operating characteristics refinement: a) Parametric diagram to compute operating characteristics curves; b) Requirement refinement through parametric diagrams.

Fig. 2-26 shows some electrical requirements derivation relationships; it can be seen that some requirements may just be copied from the initial requirements. However, with a different id, to know to which specialized package it belongs. In summary, the converter is specified; due to its specifications, the motor shall be a three-phase motor with a current less than or equal to 15 Amperes and the rated line-to-line voltage is limited to 400 V. The converter has the option of using space vector pulse-width modulation (PWM). Accordingly, a control requirement is derived stating to use this type of modulation because the operating voltage obtained from it is a maximum, thus lowering the operating current.

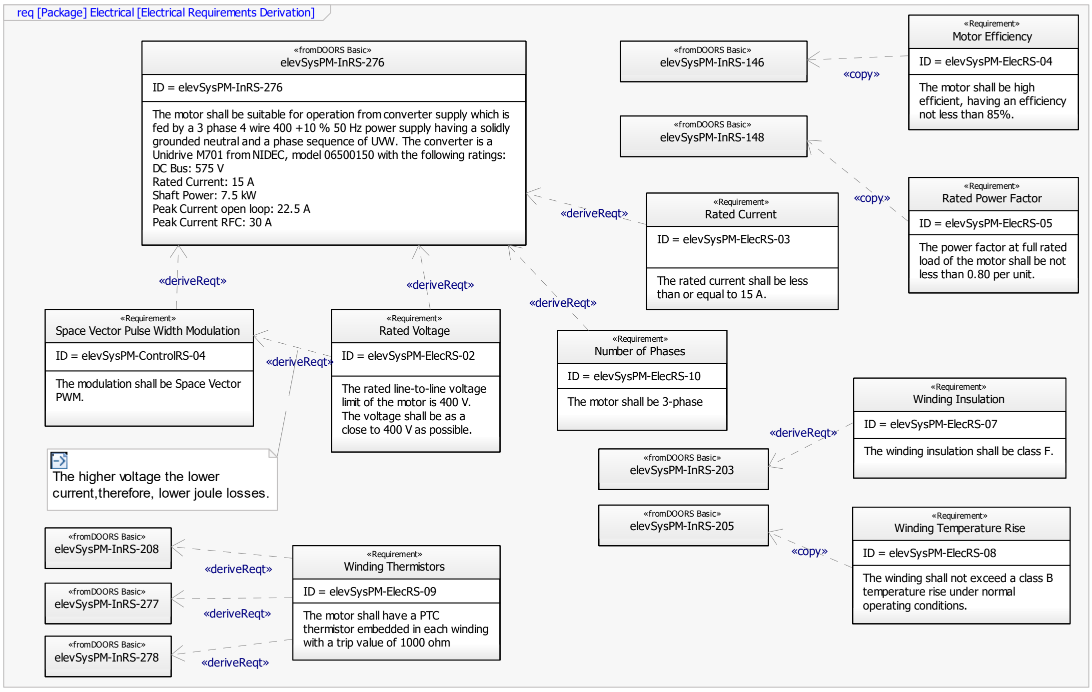


Fig. 2-26. Electrical requirements derivation.

The motor selected for this particular case is a surface permanent magnet motor, selected due to the ease of manufacturing leading to lower costs and because the required speeds are considered that can be satisfied by this type of motor.

2.4.3.3 Deriving electromagnetic requirements

This phase is reached when the aforementioned requirements are considered to be good enough by both parts to start the design phase. However, the designer/manufacturer must derive more into specific domains in order to detail better his objectives with the design. Following the framework, these requirements must be derived from the technical requirements of the different disciplines, not directly from the integrator's requirement, if the need exists, first a copy should be made into a discipline before deriving into a domain-specific package. The same process backward if the creation of a new requirement needs to go to a higher level package.

Fig. 2-27 shows an example of electromagnetic requirements derivation. In this case, the decision to use an already available design by selecting the stamping die for the stator and rotor laminations is taken based on the motor frame dimensions, type of motor, and cross-domain objective of comfort. This contains three requirements specifying the laminations for the stator, rotor, and type of steel. The stator and rotor laminations instances, also shown with its attributes in the diagram, satisfy these requirements. As a consequence of this requirement, the air gap is fixed to 1 mm. In addition, this requirement affects other domains; therefore, a copy is made to the manufacturing package. The torque ripple shall be less than 2% of the rated torque, because comfort is one of the main objectives and the requirement of the noise level. Finally, mechanical and electrical requirements, such as rated voltage, speed, torque, and others are copied.

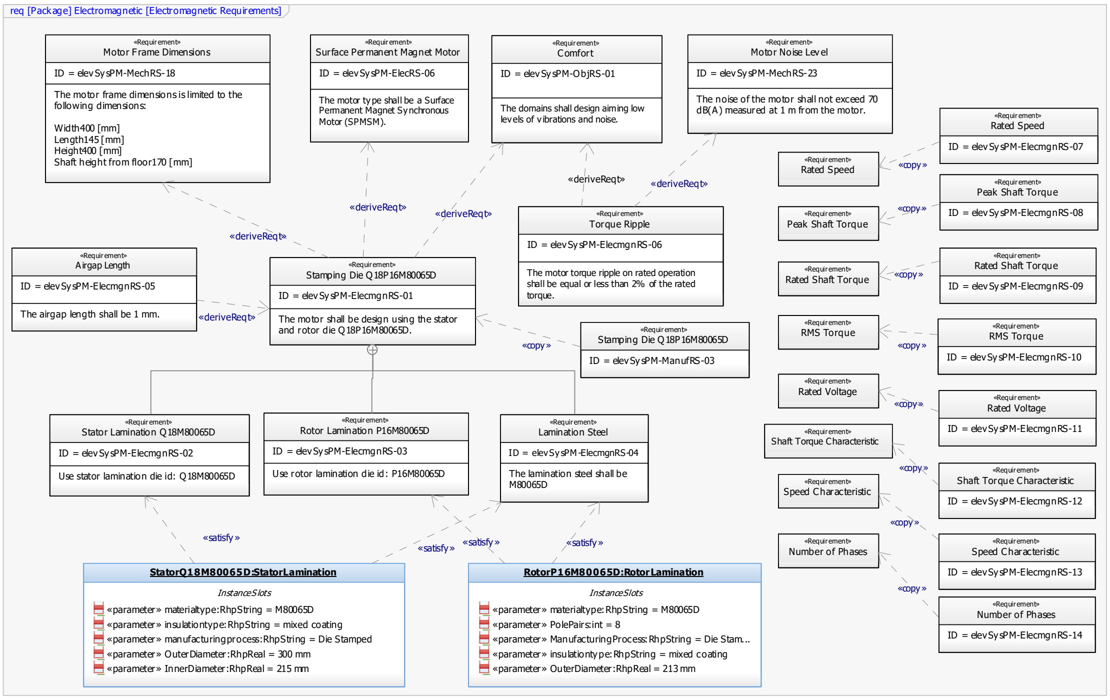


Fig. 2-27. Electromagnetic requirements derivation.

2.5 Discussion

A standardized macro-level framework for the development of electric machines was proposed. Although emphasis was put in the design process, the system also contemplates the process towards the industrialization of an electric machine considering manufacturing processes and components for serial production. The high-level system was presented with enough detail either to adapt a currently implemented development process or to use it as a starting point for a new development process. Description of tools that could be implemented and more details that complements this chapter are considered in Section 4.3.2.

Moreover, the specification stage is a critical part for the correct development of electric machines. The application of appropriate requirement engineering methods is compulsory to carry out sufficiently the process. A requirement engineering framework was presented from the perspective of the designer/manufacturer of electric machines. The framework puts special emphasis on the derivation of specialized requirements. The framework identified the stakeholders, presented a classification of requirements, and showed the activities involved in requirement analysis to end with well-defined requirements in the context of motor development. This work attempts to provide a practical guideline to the small-medium industry of electric machines to cope with the specification stage.

The requirement engineering process involves good cooperative and communication skills among the stakeholders. Therefore, it is worth outlining that this framework lacks guidelines on this communication process. The requirement framework provides the developer proper contextual means to perform the process successfully. Nevertheless, the users of the requirement engineering framework may adjust it according to their needs.

Chapter 3 KNOWLEDGE BASE GENERATION

MOKA stands for "Methodology and software tools Oriented to Knowledge based engineering Applications" [28] *p.14. It is the result of a European project (Esprit project 25418) developed with the expectation to reduce cost and time taken to develop KBE applications, which automates routine design tasks.*

MOKA philosophy is the foundation for the Knowledge Base generation process in the proposed KBES framework in this thesis, therefore, the importance of its description. The objective of this chapter is to introduce MOKA methodology and to illustrate its adaptation to the electric machine development process. In addition, the process model is extended in order to fit the purpose of representing micro-level and macro-level knowledge for electric machines.

This chapter is divided into four sections. Section one presents an introduction. Section two provides a general background of knowledge-based systems. Section three gives an overview of MOKA with a short description of the KBE lifecycle route map. Section four presents the adaptation and extension of MOKA for the electric machine development process. Finally, section five discusses the adapted methodology.

3.1 Introduction

The purpose of this chapter is to show the Knowledge Base generation process. The generation of the knowledge base is a crucial part of the KBES theory. Therefore, several methodologies have emerged to cope with this problem. Chapter 1 listed significant KBE methodologies that can be found in literature. Recently, VDI developed a new KBE guideline [109] (VDI-5610) with the intention of formulating a universally applicable process to conduct KBE projects [109], suggesting MOKA as a helpful methodology on this process. Fundamentally, any methodology could be selected. In this work is more important to select a methodology than finding the best. However, considering MOKA offers for knowledge representation the MOKA Modelling Language (MML) and a well-structured knowledge capture and formalization steps. MOKA is selected as the tool for the Knowledge Base generation process.

Nevertheless, MOKA creators intended to have a neutral methodology format and its focus is upon the capture and formalize steps. Therefore, for this thesis, an adaptation is derived regarding electric machines. In this chapter, MOKA methodology is introduced with more detail. The methodology is adapted and extended for the electric machine design process. For each step of MOKA, important insights into the development of electric machines are shared. Substantiation of the knowledge representation is presented through examples.

Curran et al. [11] identified some deficiencies of the MOKA methodology, these deficiencies are:

1. Its aim is to support the knowledge engineer instead of the end user.
2. The knowledge representation and supporting tools are not entirely identified.
3. Its target is to develop KBE applications not much attention is put to the usage in design process and does not specify in depth maintenance and knowledge reuse.
4. It is product-oriented instead of process oriented.

Thus, the adaptation of MOKA to the electric machine process was defined having in mind these deficiencies and seeking to tackle them for a better implementation process. Most of the information provided in this chapter is synthesized from the knowledge gathered from the experience of the Electric Machine Research Group from Mondragon Unibertsitatea.

3.2 Knowledge Based Systems

The purpose of this section is to clarify important concepts use in this thesis regarding Knowledge Based Systems (KBS). According to [110], Knowledge-Based Systems are software "designed to emulate the work of experts in a specific area of knowledge" to solve problems. The term usually gets confused with expert systems. Curran [29] defines an expert system as a knowledge based system application created "to offer advice, recommendations or explicit cases based on previously attained relevant knowledge to the end user".

KBS is a branch of artificial intelligence (AI) which contains expert systems, KBE, and others; this is illustrated in Fig. 3-1. Expert systems were the first knowledge based systems to be developed and received special attention because its success as first time commercial applications of artificial intelligence although because of high costs implementation and low computational capacity as well as lack of knowledge representation methodologies at the time its development did not fulfilled the product engineering design industry expectations.

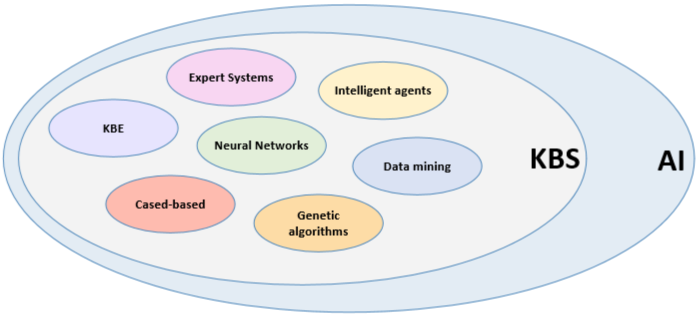


Fig. 3-1. KBS in AI domain

For the domain of KBS, common terms important in this thesis are defined in Table 3-1. Many examples of KBS can be found in different disciplines as medicine, engineering, finance, scheduling, military and others [111]–[116]. One of the pioneer expert systems is MYCIN developed by Stanford University concerning medical diagnosis and treatment of infectious diseases. A KBS consists of a knowledge base and an inference engine that are defined in the aforementioned table. To develop a KBS three kind of disciplines are involved, software engineering, domain experts and knowledge engineering [117].

KBS is suitable to be implemented in knowledge intensive tasks. Fig. 3-2 shows a typology of some of these knowledge intensive tasks as described in CommonKADS [118]. The lack of methodologies to develop KBS has been one of the major disadvantages to their implementations in engineering industry. Lately, several approaches has been developed but few offer a complete methodology with theory, methods and tools to establish a correct implementation. Chapter 1 explained that the interest of this work is KBE applied to the development process of electric machines. A literature review of KBE with current methodologies was given in Section 1.5. Finally, as explained in the introduction of this chapter MOKA has been selected and it is described in the next section.

Table 3-1. KBS related concepts. Adapted from [112], [117]–[120].

Concept	Definition
Data	"Raw digital material". [118] p. 2
Information	"Interpreted data". [118] p.2
Domain	Subject or "area of interest". [118] p.2
Knowledge	"The information about a domain that can be used to solve tasks in that domain". [117] https://artint.info/2e/html/ArtInt2e.Ch1.S4.SS2.html#p4
Knowledge base	The knowledge encoded using one of the several representation techniques, which encloses specialized knowledge on a given subject that makes a human an expert on a specific domain. [112] p.172
Knowledge engineering	The engineering discipline that deals with the process of analyzing, building and maintaining KBS involving methods, languages and tools. [120] p. 162
Heuristic knowledge	"A rule of thumb or argument derived from experience". [119] p. 2
Explicit knowledge	"Knowledge expressed in numbers/words and shared in the form of data, formulas, specifications, manuals and universals principals". [119] p. 2
Tacit knowledge	"Knowledge stored in subconscious mind of experts and not easy to document e.g. intuitions, emotions, mental models". [119] p. 2
Inference Engine	Algorithms intending to handle the knowledge inside the knowledge base similar to human reasoning with the ability to derive new information about the problem [112], [113].

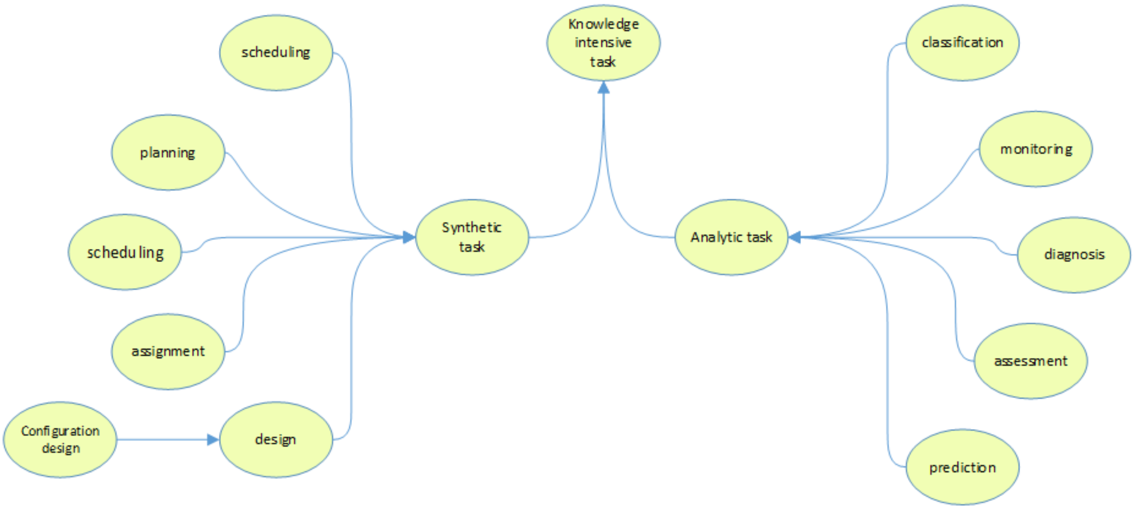


Fig. 3-2. Knowledge intensive tasks. Adapted from [118].

3.3 MOKA methodology: an overview

MOKA stands for "Methodology and software tools Oriented to Knowledge based engineering Applications" [28] p.14. MOKA methodology is the result of a European project (Esprit project 25418) and defines KBE as:

"The use of advanced software techniques to capture and re-use product and process knowledge in an integrated way" [28] p.11.

This definition is adopted throughout this document with particular emphasis on the design process. The purpose of MOKA as any other KBE methodology is to reduce time and cost for knowledge-based engineering applications concerning automation of repetitive design tasks while integrating multidisciplinary expertise domains. MOKA acceptance among industry makes it one of the most implemented KBE methodologies [6], [121]. MOKA is based on the six KBE lifecycle steps shown in Fig. 3-3. MOKA literature focus mainly on the technical aspects, the capture and formalize steps since the other steps are more application dependent.

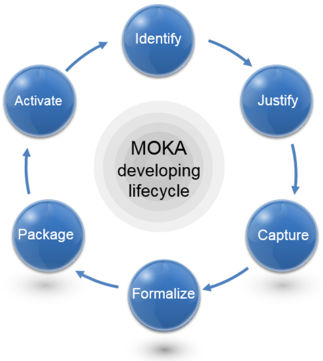


Fig. 3-3. KBE developing lifecycle of MOKA Methodology [28].

A brief description of each step is now given.

Identify: The process susceptible for KBE application are those where the knowledge is well defined and its use is repetitive. According to MOKA no KBES should be use when the design process knowledge is not defined clearly, the technology on the design process changes regularly, the design process can be modeled in a simple program, the knowledge is not available, and the organization does not want to invest in a KBES. The activities in this stage may include stating the objectives and requirements for the target KBE application, to define stakeholders, to set the scope of the application, to make sure knowledge is well known and can be captured, to define the platform where it will be implemented.

Justify: It may involve planning, costs, risk management and technical feasibility to use a KBE solution. The main objectives of this phase are estimation of resources, assessment of risks, success criteria definition, project plan, use case, win directors acceptance.

Capture: This involves the process of collecting knowledge, the sources can be experts or any other sources (e.g. documents), then to structure it into the MOKA informal model representation. To create this informal model, the knowledge engineer must use the ICARE forms (Illustration, Constraints, Activities, Rules, and Entities). The goal is to have as output a knowledge book, which is a text version of the ICARE forms with the purpose to record the expertise knowledge of the company.

Formalize: In this step, the knowledge is formalized to provide a representation without the need for formal programming skills. This formal model helps to fill the gap between the knowledge engineer and the software developer, in other words, between the informal knowledge and the computer language. The formal knowledge is divided into two models, the product model and the design process model. The MOKA consortium created the MOKA Modeling Language (MML) to represent the knowledge in both models with the flexibility to be adapted if needed. The MML is an extension of the Unified Modeling Language (UML). The product model metamodel is shown in Fig. 3-4. It is composed of five views: structure, function, behavior, technology, and representation. These views are linked together in order to provide the proper semantic description of the product.

The design process model main goal is to provide the “how” of the of the design process, this design process can be viewed from the macro-level and micro-level perspective. However, because the process is application dependent, MOKA only provides some suggestions for the micro-level view with activity diagrams. In the proposed framework, a new design process model is proposed as an extension in order to provide a clear knowledge representation. Both, the product model and design process model are customized with the purpose to make simpler the knowledge representation regarding the development of electric machines.

Package: This step involves the KBE application developing process; the application is built by following the formalized knowledge in the previous step. MOKA states that the translation of the formalized knowledge to code can be done either through a manual or automatic process, however no automatic tool is provided. For the scope of this thesis the process is manual.

Activate: This step is basically the KBE deployment phase where the KBE application is delivered to the end users, training of the tool is provided until the tool is in use.

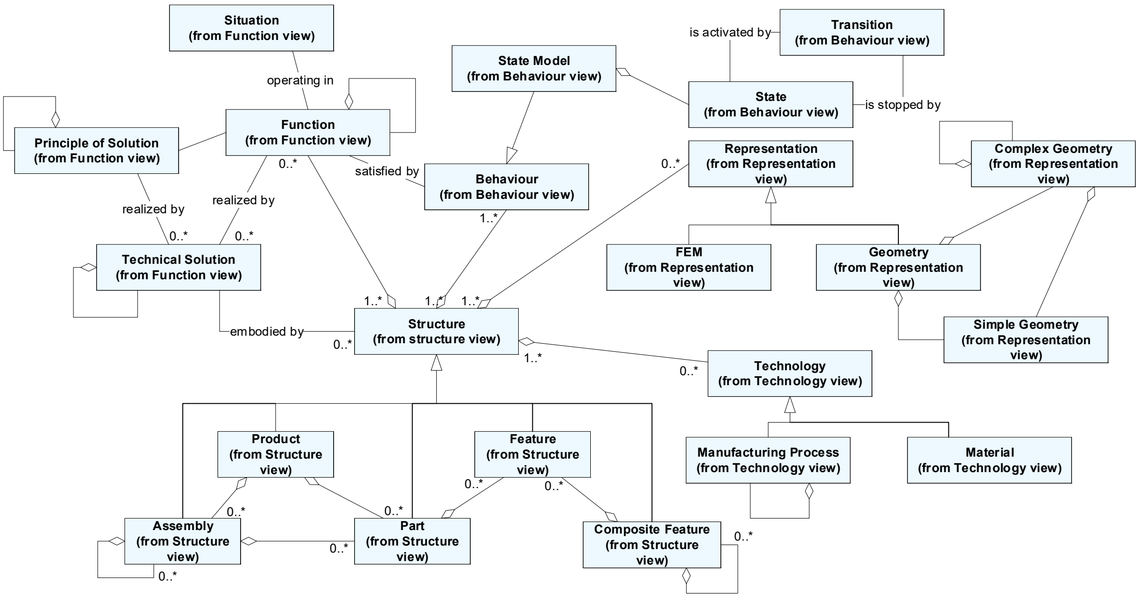


Fig. 3-4. MOKA Product metamodel [28] p.203.

MOKA generalizes the actors throughout the KBE lifecycle as: knowledge engineer, experts, developer, end-users, project managers, and general managers. The framework in this document focus only in the knowledge engineer, developer, experts, and end-users. The knowledge engineer is the actor in charge of formalizing the knowledge, process that is carried out after capturing it. The experts are one source of knowledge and are responsible to define the scope of the KBE application. The developer is the coder of the KBE application and reads or interprets the formalized knowledge. Finally, the end-user is the person using the final KBE-application. The end-user can also be an expert.

This section introduced MOKA to make the reader familiar with the methodology and its terminology. The knowledge integration process of this framework follows a similar structure as in MOKA methodology. The reader can go into more depth about MOKA in [28].

3.4 MOKA adaptation & extension applied to the design of eMachine

In order to apply the MOKA methodology to the field of electric machine design, several considerations and adaptations are needed. From the literature review, it can be said that there are no references that apply MOKA methodology to the domain of electric machines. Therefore, a new derivation has been developed in this thesis. In this section, the key aspects and the generic adaptations and extensions are presented following the steps of MOKA.

3.4.1 Identify

The first step is to identify the goals of the KBE system oriented to the real needs and limitations of the specific company. In this first step, it is important to analyze the technical viability according to the knowledge sources of the company and the development efforts that will be available. This study cannot be generic.

This work is developed with research purpose, thus, this section uses general knowledge about electric machines. The knowledge source will not be the knowledge of a particular enterprise. Hence, the knowledge sources are literature and experts on the different domains from Mondragon Unibertsitatea. These experts have experience on the design of electric machines in the realm of research and industrial applications. The standardized macro-level framework of this thesis is oriented to European SME involved in the electric machines development industry. A detailed identification could be obtained by making a study of the representative companies covering the different sectors and enterprise sizes. However, this study is out of the scope of this thesis. Then, common characteristics of these companies are obtained from the experience of the Electric Machine Research Group from Mondragon Unibertsitatea. This group has cooperated with different Basque Country SME electric machine manufacturers covering different sectors and company sizes.

Considering this information, the Identify task is performed following the next steps of MOKA methodology.

1. **Identify stakeholders and clarify their needs and objectives**

The following are potential clients identified.

Type A client: SME of the electric machine industry (main clients)

Small and medium European enterprises in the electric machines development industry, especially those having machines with a high degree of specialization in their sector (industrial applications, marine, high power, elevators, cranes, automotive, aerospace, etc.). Typically, they offer customized products to the final integrator of the machine into the application.

Needs:

The electric machinery industry is diverse having different sizes and needs. In many cases, the companies' dependence on a single sector makes its future competitiveness highly reliant on the variation of the market sales in that specific sector. To cope with this dependence they usually make plans of opening new products to new sectors and expand their businesses internationally.

They also have the need to adapt their design processes to the rapid integration of new technologies (topologies, materials, etc.) and product customization to the client in shorter periods. These enterprises are in the process of complete digitalization of their engineering and manufacturing departments, they are also seeing the advantages of using the powerful commercial CAE/CAD design software, although they still rely on their own tools to make calculations for their more conventional designs. The use of CAD/CAE software is not completely agile, because it requires significant engineering resources for its implementation. Normally, few specialized technical personnel manage these CAD/CAE tools, which may lead to risks because of delays in simultaneous design works due to lack of more trained personnel.

In summary, it can be said that the KBE system needs are:

- A comprehensive structure that enables innovation, standardization of manufacturing-oriented design, customization of the product, and a rapid response to new developments.
- Investment adjusted to features and real needs of the enterprise. Investment adjusted to the size and requirements of new products.
- Integration of current engineering processes and good practices of the enterprise to facilitate the transition and make faster the new system implementation.

Objectives:

The devised KBE system for this type of client would have the following objectives.

- Systematize and leverage the complete knowledge of the company to improve innovation in designs and shorter development times.
- Use advanced design tools (own and commercial) to ensure reliability and accurate computation in the different design stages.
- Use of standardized processes for either automation or manual execution.
- Provide support to the end-user (designer, supplier, client, manager, etc.) with routine tasks automation. Opening new advanced capabilities associated with the intelligent use of information (search for proper solutions to problems, warnings of design parameters out of range, intelligent assistance, etc.).

Type B client: Technological agents (secondary clients):

Nowadays, technological agents (technology centers, universities, research labs, etc.) collaborate with European electric machine companies to provide technological and research support. The support can be given directly to the enterprise or through public European or state research organizations. The design system of these agents could be part of the global system of the enterprise to which they provide support or can be managed independently.

Needs:

The design systems of technological agents are focused in the search for innovative solutions. Their efforts are based in the development of own advanced design tools and use of cutting-edge commercial CAD/CAE software. Typically, they are not sensitive to engineering efforts for new design customization nor to the development of low cost manufacturing concepts (i.e. reuse of manufacturing resources of the company). The agents give solutions to needs in different sectors and there is no guarantee of continuous work in one particular sector. They cannot afford to make big research efforts for a specific sector.

In summary, for this client the KBE system needs are:

- A system that provides innovation capabilities, and is compatible with the systems of the companies with which it works.
- Investment in generic features of the system that can give support for different technologies of electric machines and industrial applications. Flexible and scalable system that can be progressively expanded needing low initial investments.
- The system should considered the final manufacturing processes that will allow building a cost effective electric machine.

Objectives:

The devised KBE system could have similar objectives as the type A client but with the following differences:

- Use of knowledge from own innovations and recent research from technologies and scientific domains to guarantee a wide design space.
- Provide fast integration of the latest commercial design tools. Collaboration with CAE/CAD SW developers can facilitate the integration of own developed tools into their software or their design tools into the KBE application. The objective is to be at the forefront in high fidelity computation of the different physical phenomena in electric machines.
- Follow standardized processes. Although within each micro-process, the system can give freedom to the designer to apply its own ideas without constraints.

- Incorporate artificial intelligence whenever possible to make results predictions or find design patterns for different cases by data analysis.

2. Define the role and scope of possible application of the KBE system

In this case, it is important to define the main value added that the company seek in the development process. In some cases, the manufacturing company prefers to outsource R&D efforts, and prefers to focus its objectives on having a product customization based on limited product solutions in which it is ensured a cost effective manufacturing. For these cases, it is more advisable that the KBE system is center on the Industrial V-cycle. On the other hand, an innovation laboratory could be focus on the Concept V-cycle. Regardless, this work proposes that although there are different levels of automation that can be obtained both SME and technological centers should integrate the three V-cycles.

3. Identify knowledge sources and the ability to acquire it

In this phase, the knowledge sources regarding design and manufacturing of electric machines are defined. These knowledge sources are characterized in types such as persons, documents, models, software. At the same time, the ability to acquire the knowledge is evaluated. For instance, knowledge can also be in an external company but could also be captured if it has part of the internal know-how through mutual collaboration.

In general terms, for the electric machines industry the following knowledge sources have been identified.

- **Personnel**: Experienced design and manufacturing technicians are a source of implicit knowledge. These personnel have solved most of the engineering problems the enterprise has faced. For the case of electric machines some problems are of high theoretical complexity, many of these problems have been solved in practical ways by applying learned engineering methods (e.g. optimal combination of number of slots and poles for different cases like reducing vibro-acoustic problems). Although this knowledge is accessible and MOKA offers processes to obtain this informal knowledge, the challenge for the company is on finding the methods to obtain it when the personnel knows is for a KBE system. This means that the company should guarantee a good social-work environment among the employees when working on obtaining their knowledge. In this context, an employee may feel threatened with losing his job due to the KBE system, hence, he will possibly have not a good attitude towards giving his acquired knowledge.
- **Technical documents**: This knowledge source is quantifiable, expressed in technical language as rules (equations, formulas, algorithms etc.), simulation results, testing results, literature

references. The process to extract this knowledge can be focus on the analysis of these documents by the expert in the domain and capture it with the support of the knowledge engineer through ICARE forms or directly formalize it through language representation.

- **Engineering computation tools**: Within each company there are internal developed computation tools (implemented either through software or manual operation) that has been validated throughout the experience of the enterprise. The knowledge is implicit if the context of the rules used within these tools is not kept in a knowledge base. However, this type of knowledge is defined in a deterministic way through the algorithms employed and the processes of computation.
- **Manufacturing and quality assurance processes**: In an enterprise a series of manufacturing and quality assurance processes are undertaken to guarantee the offering of a competitive product with the best price-quality relation. Normally, these processes have been optimized accounting the history of incidents and continuous improvement programs. The whole set of these processes represent a true intangible asset for the company. The systematization of this knowledge through the KBE technology offers opportunities to get the most out of the know-how when developing new products.
- **Prototypes**: In this case, the knowledge is expressed as a prototype or a real manufactured product that fulfills a set of specifications and experimental validated performance. If the requirements specification, design computation process, and experimental validated results were available, this information would be a good source of knowledge. In addition, it can be included the followed manufacturing layouts, manufacturing processes and mounting. The maintenance manuals are also a source worth of consideration. The physical prototype is a source of direct knowledge by extracting information on how the design process has been implemented. But could also be an indirect source of knowledge by having to make tests to the prototype to validate different models of computations for the different multiphysics domains.
- **Suppliers**: At the beginning of the manufacturing value chain of electric machines, we have the suppliers of raw materials and manufacturing tooling. These suppliers contain knowledge specialized in their products. The know-how contained by these suppliers typically is within their R&D departments. In many occasions, they need their customers employ their new products or new manufacturing methods in order to validate them in developed electric machines. For this reason, there is an open opportunity for electric machines manufacturers to carry out joint development work and to acquire from the suppliers significant knowledge of new materials and manufacturing processes. It is important to remark, that there are sensitive aspects of the knowledge of the suppliers that they might not be willing to share since it is a core part of their competitiveness. Therefore, these aspects should be identified to well determine the scope of the

joint collaboration. For example, the material supplier can have as confidential the internal manufacturing process of the material but can share details of the properties of the material to optimize its use in the design and manufacturing of electric machines. Another example could be that die manufacturers may not give you information of the process of tooling of your die but can share the tolerances of the output cut in the sheets for consideration in design. Respecting the sensitive know-how of suppliers may open them to share in short or long term their knowledge that will enrich current development of electric machines.

- **Technological suppliers**: Technology suppliers are agents that their contract is limited to offer a specific problem solution. Similar to the previous suppliers these can be an important source of knowledge because they are specialized in a specific scientific and/or technological field. The knowledge is withheld within the project deliverables associated to the collaboration contract (drawings, reports, demonstrations, etc.). Regarding the devised KBE system, it is convenient to define the scope and format of the deliverables taking into account its inclusion in the knowledge base. An alternative scheme of the relationship supplier/manufacturer could be based in a long-term collaboration where the technological supplier personnel is a knowledge source for the manufacturing enterprise. In this case, besides the deliverables for specific contracts the personnel can be a treated as internal employees from which knowledge can be gathered. For this purpose, there should be intellectual property agreements between both parties. It is worth to highlight that the formalization of knowledge for a KBE system makes explicit what knowledge is shared. Making possible to engage into clear agreements avoiding misunderstandings in the scope of the know-how shared.
- **Clients**: Within the development scheme of an optimized electric machine as end product for a particular application. The client has the information about the needs of the application. Therefore, they are the source of knowledge to know what requirements and KPI are important. The requirements specification itself can be used to formalize the knowledge for specific applications. Generalizing these requirements through knowledge engineering methods may provide global knowledge giving the possibility of anticipating variants of electric machines for an application. Furthermore, a mutual collaboration between client/manufacturer can lead to the generation of knowledge sources that can predict future needs of the clients becoming in future developments and improvements to electric machines.
- **Competitive monitoring and technological/scientific state of the art**: Analysis of patents, product competitive benchmarking, and keeping track of scientific and technological state of the art in electric machines represent an important source of knowledge that a company should contemplate. The enterprise should evaluate its resources to perform proper competitive surveillance. In case its resources are limited it is advisable alliances with other enterprises to

share resources to cover all global references that are related to their product seeking improvements in the market.

This thesis has based his knowledge sources mainly on the R&D team of Mondragon Unibertsitatea Group of Electric Machines. It is considered that due to the experience of this team, a sufficient source of knowledge has been obtained to show the potential of a KBE system. It is not an objective of this work to define general or specific knowledge for each type of electric machine developer. Instead, the goal is to show a representative knowledge base and the process of its generation to begin a first concept implementation of a KBE system for a future practical application for an electric machine developer.

4. Identify means of knowledge capture

Depending on the type of knowledge source, the means to capture this knowledge will be identified. It is important to evaluate the time and manner of capturing information from expert designers. For the process to be effective, the time should be short and the complexity required low otherwise it is likely the capture process will be tedious and not properly executed.

Next, a first proposal of knowledge capture means is presented for the different knowledge sources.

- **Personnel**: In this case, the knowledge is informal or tacit. The means of capturing it can be through formal interviews conducted by a knowledge engineer to fill in the ICARE forms. Likewise, in addition to the knowledge engineer it would be helpful to have an engineer that knows the domain not necessarily an expert but that has skills on integrating technologies and processes for instance a systems engineer. Eventually, in a process of continuous improvement the same personnel will be able to fill out the ICARE forms by themselves.
- **Technical documents**: The knowledge is explicit expressed in technical language supported by equations, simulation results and literature references. The means of capturing this knowledge will be through depth analysis to extract the relevant knowledge to the KBE system. The expert should carry out the process supported by the knowledge engineer who will formalize it.
- **Engineering computation tools**: To use a developed computation tool effectively, you must first ensure that its results are reliable. Therefore, the complete process should be validated: problem definition, proper parameter selection, inputs and certainty of results interpretation. If the process is written in manuals, the knowledge should be captured using the process as with technical documents. If the process is within the expertise of the developer then the knowledge should be captured following the process mentioned in personnel.

Once the tool is validated (reliable and accurate) then the tool is a valuable source of knowledge. There are two ways to use this source of knowledge:

a) As explicit knowledge: the tools represent explicit form of the physical phenomena (e.g. analytical equations) or technical solutions (e.g. diagrams of solutions extracted from a database). The domain expert can extract this knowledge to save it in generalized form in the knowledge base. From this knowledge other solutions may arise, that is why it is convenient to have it in a knowledge base in order to facilitate the reuse of knowledge in other tools or other processes.
b) As implicit knowledge: If there is no access to the knowledge within the tool perhaps because of its complexity or is a blackbox and only the relation input/outputs can be used. Then the tool can be used directly as a solution to support a particular process in the design phases. Another usage could be to make a sensitivity analysis correlating inputs and internal parameters to see the output behavior to these variations. Finally, it can also be used as a tool to validate explicit analytical equations for a problem in a bounded operating environment.

- **Prototype**: Additionally to the documents of a prototype, the physical prototype or product is a knowledge source. The knowledge capture means can be performance tests, stress tests, accelerated age tests, etc. Likewise, the direct analysis of mechanical tolerances between the different parts, the mechanical properties of the parts such as if they have suffered corrosion, wear, micro-tears, etc. are means to understand and know the health of the electric machine subjected to a certain lifecycle. In the last phase of life of the prototype, the parts can be dismounted making possible the analysis of degradation of parts through external dimensions measurement, micro-tear analysis in microscope as well as chemical properties analysis of materials and mechanical tests.

5. Identify target KBE platforms

Here is undertaken the evaluation of the degree of particularity required for the KBE platform and if commercial tools can be integrated to fulfill objectives.

In this thesis, several tools have been analyzed. Knowledge formalization tools such as Protégé, MML, PROLOG, LISP, SysML, and others. Product data management (PDM) and product lifecycle management (PLM) tools, requirement management tools, design tools for specific domains and generic optimization tools. While each tool alone does not meet the needs of a KBE system for the company, herein however, it is considered the intelligent and systematic use of these tools through a customized framework for a company can reduce the workload and costs of an implementation of a particular KBE system developed from scratch. Furthermore, not only the implementation should be taken into account but also the maintenance, continuous improvements and expansion of the KBE system. When choosing a commercial or own developed platform it should be flexible enough to adapt to new versions and integration of new tools.

If the KBE system is limited in terms of tasks to execute and complexity, the enterprise itself could implement it, avoiding licenses costs of external software. Nevertheless, taking into account the maintenance of the platform will be done by the enterprise.

In conclusion, it can be said that current commercial software can become part of the devised KBE system to provide particular solutions. These software platforms even have the capability of interconnection among third party tools and automatic information sharing. Therefore, it exist the possibility to develop a centralized software tool to command all the integrated tools. The use of commercial tools allows a faster implementation and improvements of functionalities since each software supplier offer better capabilities in newer versions of their product. However, it is worth considering that there is a dependence with external software suppliers implicating license cost, low flexibility in internal changes in their tools, vulnerable to external decisions and versions updates of the software (e.g. lack of compatibility of newer versions and discontinue software). On the other hand, an implementation by the enterprise itself requires limiting the scope of the KBE system, management of economic and human resources for its implementation and maintenance.

6. **Assess technical feasibility**

Taking into account the aforementioned points, the technical evaluation of knowledge assistance support to the design process is analyzed. It is a challenge to accomplish a KBE system that includes multidisciplinary knowledge intensive design processes. In some cases, even the end goal of the KBE system can be lost during its implementation, which is to make it a useful tool to support designers. Some parts of the design process could be complex to formalize its knowledge making the formalization process efforts larger than the contribution it might give. Then, for these cases would be more practical to represent it in an informal way linking the knowledge to its original source (people, computer software, drawings, etc.).

3.4.2 Justify

Following MOKA methodology, this task have the next steps.

1. **Estimate resource requirements and costs**

For each devised KBE system the cost estimation of implementation as well as estimation of future necessary resources should be carry out.

2. **Assess technical, cultural and commercial risks**

The business culture should be taken into account to define the degree of acceptance of an enterprise to a KBE system.

3. **Define acceptance criteria**

For each envisage KBE system the definition of what criteria can make the project acceptable to the stakeholders of the enterprise should be analyzed. With special emphasis in the stakeholders that intervene in the design, manufacturing and validation of the electric machines.

4. **Generate a project plan**

In this stage, future projects using the KBE system should be defined including the different scenarios and products.

5. **Prepare the business case**

For each project, a business plan would be defined. It is important to take into account the different actors involved in the design process of a new electric machine. Some manufacturers choose to outsource a large part of the design process, therefore, the role of external actors and how they participate both technically and financially in the integral KBE system should be taken into account.

6. **Gain management approval**

In this case, the necessary information to be shared between the internal / external provider of the KBE system and the internal stakeholders of the company will be defined to make the project be accepted in a solid way.

This thesis centers its attention on the technical engineering aspects of KBE, administrative factors are not considered in depth although mentioned throughout the work. The administrative part is considered to be closely linked to the complete structure of each company. In general, the main activity is to capture and formalize the knowledge, to make later a progressive implementation of the KBE system.

Likewise, another business model not directly contemplated in the thesis could come from a generic software company that would commercialize a generic KBE package that could be distributed to the different electric machine manufacturers. It should be noted that the needs of different companies are very different, and making a very complete generic KBE system would require great development efforts and the final acceptance by the manufacturer of the electric machine might have risks; since companies are willing to pay exclusively for the design assistance features that they specifically need.

From the analysis of commercial multi-domain software products associated with mechatronic systems, an integration of commercial programs has been observed through a series of SW business purchases. Thus, there are a small number of technology providers (Ansys, Altair, Siemens, etc.) that offer a suite of tools for each domain (increasing compatibility with each other) to be used by the end customer in a simpler and more comprehensive way. These comprehensive tools are offering multi-purpose generic designer assistance modules, initially more focused on optimizing the calculation of the specific domain rather than a vertical extension of the entire design process.

It is recommended to manufacturers of electric machinery to make continuous surveillance of the state of the art and to extend the collaboration with these technological suppliers of specialized SW multi-domain. From the capabilities of these companies to offer specialized design assistance products; it has been observed that the proposals in the market focus more on generalized products that can be used by a large number of clients (universities, manufacturers of various sectors) rather than products providing an integral response to a particular sector for the development of electric machines. Analyzing the future perspective regarding KBE systems, electric machines developers can integrate these platforms of commercial SW because the SW by themselves can hardly give a complete solution to the specific company needs.

3.4.3 Capture

This stage covers the raw knowledge collection and structure of the knowledge by means of the ICARE forms (Illustration, Constraints, Activity, Rules and Entity forms). The collection of this forms create the informal model. The main idea is to have a coherent knowledge representation understandable by expert designers and the knowledge engineer facilitating them to work together. The result of this phase is a knowledge book withholding the knowledge.

The steps to follow in the capture process are:

1. **Prepare for collection**

In this step, the assurance that the previously identified knowledge is relevant is carried out. In addition, that the knowledge is accessible and a capture plan is prepared with defined procedures and means of storage of information.

2. **Collect required knowledge**

This task will capture the knowledge of design experts of electric machines and reference documents. The method to capture the knowledge is mainly through an on-site registration of the process and considerations that the expert follows in carrying out each process of the V-cycle. Regarding the documents information, a structure technique of categorizing the proposed information will followed as recommended in MOKA.

3. **Structure the raw knowledge**

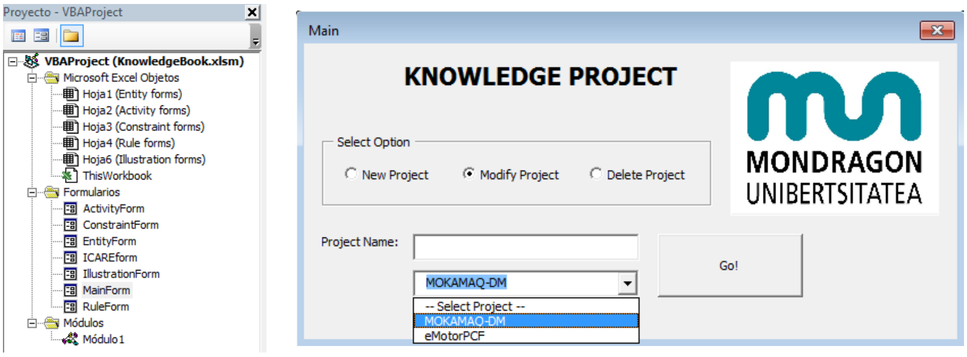
In this step, the raw knowledge is structured filling out MOKA ICARE forms. The knowledge is detailed as entity (parents, child), illustration, constraint, activity, and rules.

4. Check fitness for purpose

During this step, a revision is done assuring that the structured knowledge matches the initial gathered knowledge. To support this, several use case examples can be analyzed and the process is studied and compared to both the knowledge source and structured knowledge.

5. Annotate and file models

The gathered knowledge is stored in a repository and it represents a readable source of informal knowledge by any designer. In this thesis, a generic knowledge structure about the design of the electric machines is generated. To create the knowledge book and simplify the task of knowledge capture, a macro with the ICARE forms was developed using Visual basic in Microsoft Excel.



Select Option: ○ New Project ● Modify Project ○ Delete Project

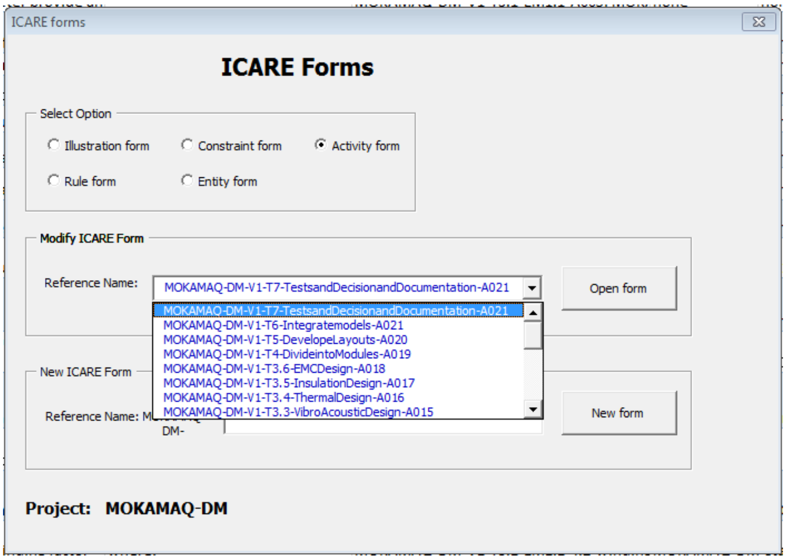
MOKAMAQ-DM

a) b)

Fig. 3-5. a) Macro creation; b) Main window: access to a knowledge project.

Fig. 3-5a. Fig. 3-5b shows the main window, it gives the options of creating a new project, modifying (or continuing) an existing project or deleting a project.

Fig. 3-6 shows the following window where the user can opt for any of the ICARE forms, permitting to either modify (or visualize) or create a new one.



Select Option: ○ Illustration form ○ Constraint form ● Activity form ○ Rule form ○ Entity form

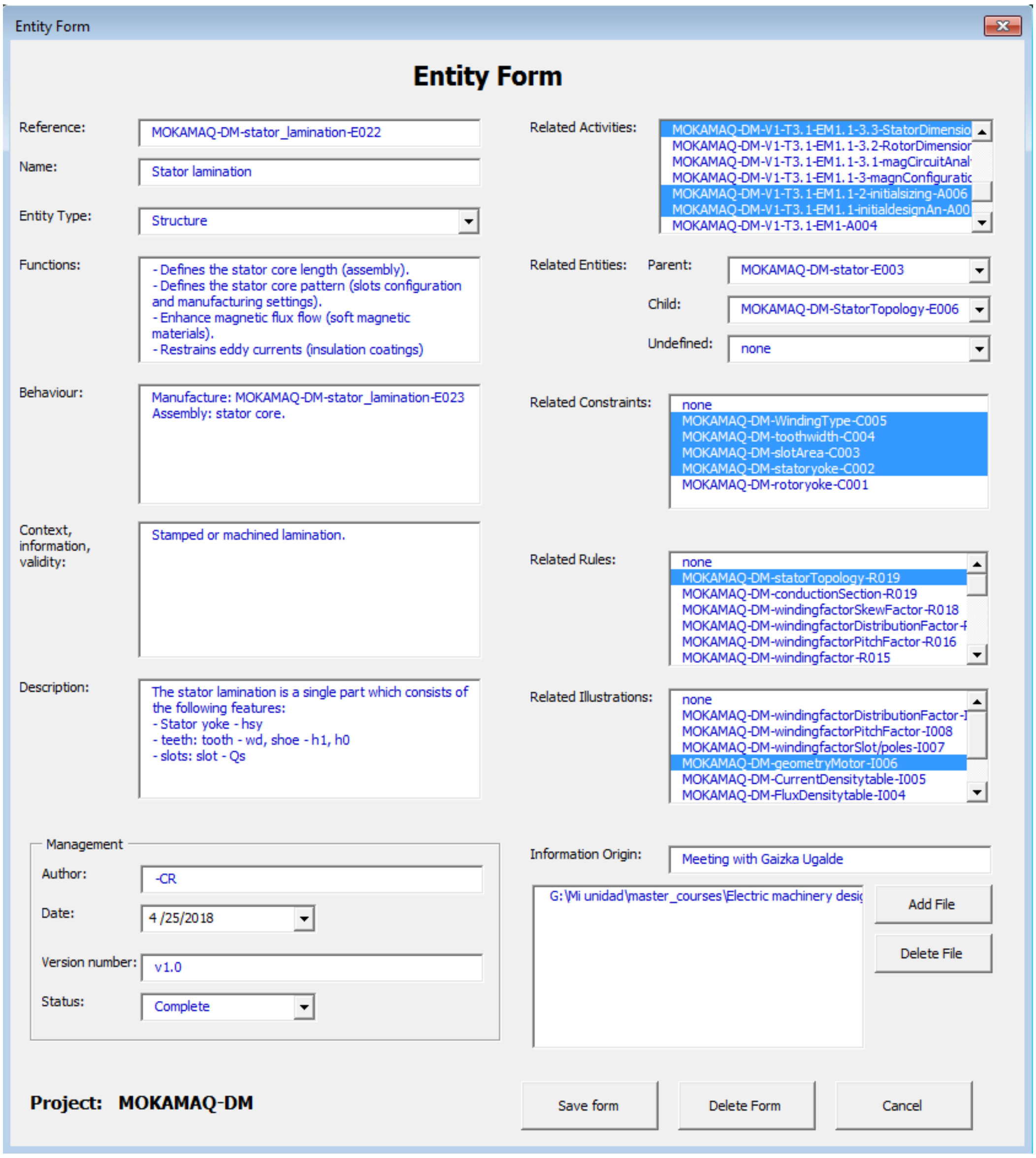
Reference Name: MOKAMAQ-DM-V1-T7-TestsandDecisionandDocumentation-A021

MOKAMAQ-DM-V1-T7-TestsandDecisionandDocumentation-A021
MOKAMAQ-DM-V1-T6-Integratemodels-A021
MOKAMAQ-DM-V1-T5-DevelopeLayouts-A020
MOKAMAQ-DM-V1-T4-DivideintoModules-A019
MOKAMAQ-DM-V1-T3.6-EMCDesign-A018
MOKAMAQ-DM-V1-T3.5-InsulationDesign-A017
MOKAMAQ-DM-V1-T3.4-ThermalDesign-A016
MOKAMAQ-DM-V1-T3.3-VibroAcousticDesign-A015

Project: MOKAMAQ-DM

Fig. 3-6. ICARE form selection.

Fig. 3-7 shows as an ICARE form example, one rule-form with all the fields. In this case, the form shows the entity form of the stator lamination with its functions, related manufacturing processes, features and other forms linked to this one in order to understand the context of this part.



Entity Form

Reference: MOKAMAQ-DM-stator_lamination-E022

Name: Stator lamination

Entity Type: Structure

Functions:
- Defines the stator core length (assembly).
- Defines the stator core pattern (slots configuration and manufacturing settings).
- Enhance magnetic flux flow (soft magnetic materials).
- Restrains eddy currents (insulation coatings)

Behaviour:
Manufacture: MOKAMAQ-DM-stator_lamination-E023
Assembly: stator core.

Context, information, validity: Stamped or machined lamination.

Description:
The stator lamination is a single part which consists of the following features:
- Stator yoke - hsy
- teeth: tooth - wd, shoe - h1, h0
- slots: slot - Qs

Related Activities:
MOKAMAQ-DM-V1-T3.1-EM1.1-3.3-StatorDimensio
MOKAMAQ-DM-V1-T3.1-EM1.1-3.2-RotorDimensior
MOKAMAQ-DM-V1-T3.1-EM1.1-3.1-magCircuitAnal
MOKAMAQ-DM-V1-T3.1-EM1.1-3-magnConfiguratic
MOKAMAQ-DM-V1-T3.1-EM1.1-2-initialsizing-A006
MOKAMAQ-DM-V1-T3.1-EM1.1-initialdesignAn-A00
MOKAMAQ-DM-V1-T3.1-EM1-A004

Related Entities:
Parent: MOKAMAQ-DM-stator-E003
Child: MOKAMAQ-DM-StatorTopology-E006
Undefined: none

Related Constraints:
none
MOKAMAQ-DM-WindingType-C005
MOKAMAQ-DM-toothwidth-C004
MOKAMAQ-DM-slotArea-C003
MOKAMAQ-DM-statoryoke-C002
MOKAMAQ-DM-rotoryoke-C001

Related Rules:
none
MOKAMAQ-DM-statorTopology-R019
MOKAMAQ-DM-conductionSection-R019
MOKAMAQ-DM-windingfactorSkewFactor-R018
MOKAMAQ-DM-windingfactorDistributionFactor-
MOKAMAQ-DM-windingfactorPitchFactor-R016
MOKAMAQ-DM-windingfactor-R015

Related Illustrations:
none
MOKAMAQ-DM-windingfactorDistributionFactor-
MOKAMAQ-DM-windingfactorPitchFactor-I008
MOKAMAQ-DM-windingfactorSlot/poles-I007
MOKAMAQ-DM-geometryMotor-I006
MOKAMAQ-DM-CurrentDensitytable-I005
MOKAMAQ-DM-FluxDensitytable-I004

Management
Author: -CR
Date: 4 /25/2018
Version number: v1.0
Status: Complete

Information Origin: Meeting with Gaizka Ugalde

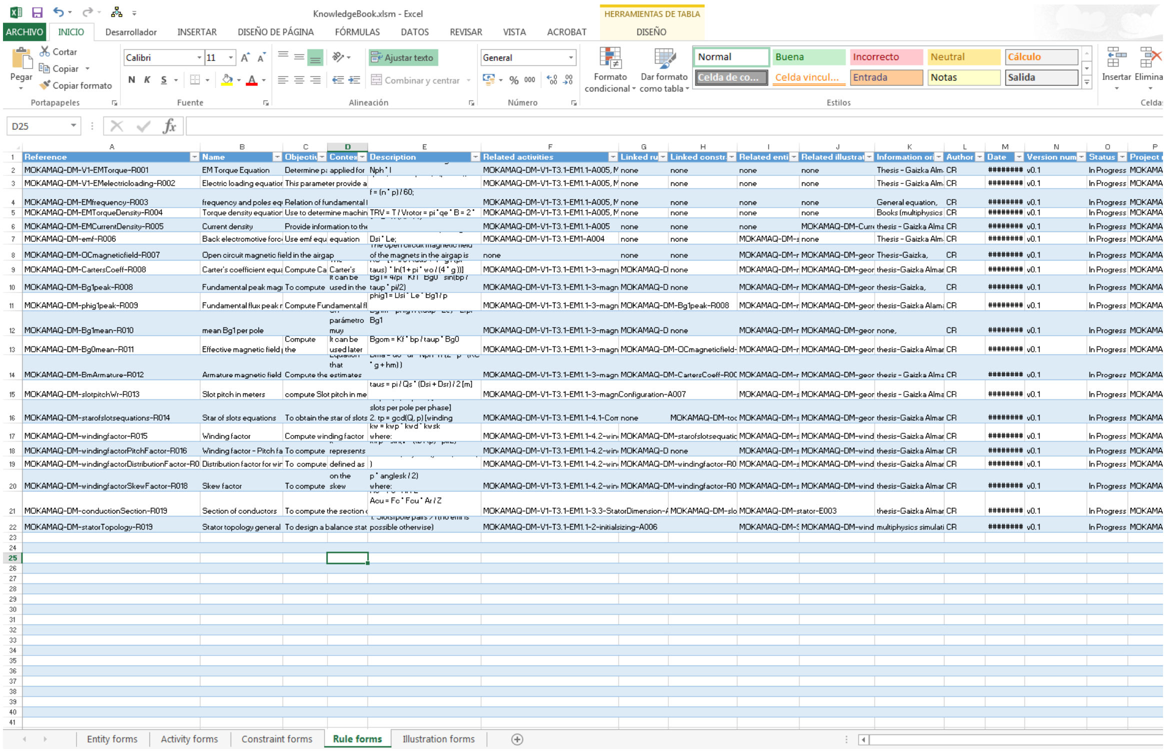
G:\Mi unidad\master_courses\Electric machinery desi

Project: MOKAMAQ-DM

Fig. 3-7. MOKA ICARE forms – Rule form.

When the user saves the form, the information gets stored automatically in its respective worksheet. At the end of the process, the result is a knowledge book. This favors the fast implementation of the capture phase and allows having the information structured as a database, which support fast filtering and modification of the data.

Fig. 3-8 presents a view of the knowledge book, a worksheet for each type of form. In this phase, the information does not reflect well the knowledge therefore the importance of the formalization step.



	A	B	C	D	E
1	Reference	Name	Objectiv	Contex	Description
2	MOKAMAQ-DM-V1-EMTorque-R001	EM Torque Equation	Determine pa	applied for	Nph * I
3	MOKAMAQ-DM-V1-EMelectricloading-R002	Electric loading equation	This parameter provide a		
4	MOKAMAQ-DM-EMfrequency-R003	frequency and poles equ	Relation of fundamental f		f = (n * p) / 60;
5	MOKAMAQ-DM-EMTorqueDensity-R004	Torque density equation	Use to determine machin		TRV = T / Vrotor = pi * qe * B = 2 *
6	MOKAMAQ-DM-EMCurrentDensity-R005	Current density	Provide information to the		
7	MOKAMAQ-DM-emf-R006	Back electromotive force	Use emf equ	equation	Dsi * Le;
8	MOKAMAQ-DM-OCmagneticfield-R007	Open circuit magnetic field in the airgap			The open circuit magnetic field of the magnets in the airgap is
9	MOKAMAQ-DM-CartersCoeff-R008	Carter's coefficient equa	Compute Ca	Carter's	taus) * ln(1 + pi * wo / (4 * g)))
10	MOKAMAQ-DM-Bg1peak-R008	Fundamental peak magn	To compute	It can be used in the	Bg1 = 4/pi Kf1 Bg0 sin(bp / taup * pi/2)
11	MOKAMAQ-DM-phig1peak-R009	Fundamental flux peak r	Compute Fundamental fl		phig1 = Dsi * Le * Bg1 / p
12	MOKAMAQ-DM-Bg1mean-R010	mean Bg1 per pole		parámetro muy	Bg1
13	MOKAMAQ-DM-Bg0mean-R011	Effective magnetic field p	Compute the	It can be used later	Bgom = Kf * bp / taup * Bg0
14	MOKAMAQ-DM-BmArmature-R012	Armature magnetic field	Compute the	that estimates	* g + hm))
15	MOKAMAQ-DM-slotpitchWr-R013	Slot pitch in meters	compute Slot pitch in me		taus = pi / Qs * (Dsi + Dsr) / 2 [m]
16	MOKAMAQ-DM-starofslotsequations-R014	Star of slots equations	To obtain the star of slots		slots per pole per phase] 2. tp = gcd(Q, p) [winding
17	MOKAMAQ-DM-windingfactor-R015	Winding factor	Compute winding factor		kw = kwp * kwd * kwsk where:
18	MOKAMAQ-DM-windingfactorPitchFactor-R016	Winding factor - Pitch fa	To compute	represents	
19	MOKAMAQ-DM-windingfactorDistributionFactor-R0	Distribution factor for wir	To compute	defined as	)
20	MOKAMAQ-DM-windingfactorSkewFactor-R018	Skew factor	To compute	on the skew	p * anglesk / 2) where:
21	MOKAMAQ-DM-conductionSection-R019	Section of conductors	To compute the section c		Acu = Fc * Fcu * Ar / Z
22	MOKAMAQ-DM-statorTopology-R019	Stator topology general	To design a balance stat		1. Slots/pole pairs > 1 (no emf is possible otherwise)

	F	G	H	I	J	K	L	M	N	O	P
1	Related activities	Linked ru	Linked constr	Related enti	Related illustrat	Information ori	Author	Date	Version num	Status	Project
2	MOKAMAQ-DM-V1-T3.1-EM1.1-A005, M	none	none	none	none	Thesis - Gaizka Alm	CR	########	v0.1	In Progress	MOKAMA
3	MOKAMAQ-DM-V1-T3.1-EM1.1-A005, M	none	none	none	none	Thesis - Gaizka Alm	CR	########	v0.1	In Progress	MOKAMA
4	MOKAMAQ-DM-V1-T3.1-EM1.1-A005, M	none	none	none	none	General equation,	CR	########	v0.1	In Progress	MOKAMA
5	MOKAMAQ-DM-V1-T3.1-EM1.1-A005, M	none	none	none	none	Books (multiphysics	CR	########	v0.1	In Progress	MOKAMA
6	MOKAMAQ-DM-V1-T3.1-EM1.1-A005	none	none	none	MOKAMAQ-DM-Curr	thesis - Gaizka Alma	CR	########	v0.1	In Progress	MOKAMA
7	MOKAMAQ-DM-V1-T3.1-EM1-A004	none	none	MOKAMAQ-DM-s	none	Thesis - Gaizka Alm	CR	########	v0.1	In Progress	MOKAMA
8	none	none	none	MOKAMAQ-DM-r	MOKAMAQ-DM-geor	Thesis-Gaizka,	CR	########	v0.1	In Progress	MOKAMA
9	MOKAMAQ-DM-V1-T3.1-EM1.1-3-magn	MOKAMAQ-D	none	MOKAMAQ-DM-r	MOKAMAQ-DM-geor	thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
10	MOKAMAQ-DM-V1-T3.1-EM1.1-3-magn	MOKAMAQ-D	none	MOKAMAQ-DM-r	MOKAMAQ-DM-geor	thesis-Gaizka,	CR	########	v0.1	In Progress	MOKAMA
11	MOKAMAQ-DM-V1-T3.1-EM1.1-3-magn	MOKAMAQ-DM-Bg1peak-R008		MOKAMAQ-DM-r	MOKAMAQ-DM-geor	thesis-Gaizka Alam	CR	########	v0.1	In Progress	MOKAMA
12	MOKAMAQ-DM-V1-T3.1-EM1.1-3-magn	MOKAMAQ-D	none	MOKAMAQ-DM-r	MOKAMAQ-DM-geor	none,	CR	########	v0.1	In Progress	MOKAMA
13	MOKAMAQ-DM-V1-T3.1-EM1.1-3-magn	MOKAMAQ-DM-OCmagneticfield-		MOKAMAQ-DM-r	MOKAMAQ-DM-geor	thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
14	MOKAMAQ-DM-V1-T3.1-EM1.1-3-magn	MOKAMAQ-DM-CartersCoeff-R0C		MOKAMAQ-DM-r	MOKAMAQ-DM-geor	thesis - Gaizka Alma	CR	########	v0.1	In Progress	MOKAMA
15	MOKAMAQ-DM-V1-T3.1-EM1.1-3-magnConfiguration-A007			MOKAMAQ-DM-s	MOKAMAQ-DM-geor	thesis - Gaizka Alma	CR	########	v0.1	In Progress	MOKAMA
16	MOKAMAQ-DM-V1-T3.1-EM1.1-4.1-Con	none	MOKAMAQ-DM-toc	MOKAMAQ-DM-s	MOKAMAQ-DM-geor	thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
17	MOKAMAQ-DM-V1-T3.1-EM1.1-4.2-win	MOKAMAQ-DM-starofslotsequatic		MOKAMAQ-DM-s	MOKAMAQ-DM-wind	thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
18	MOKAMAQ-DM-V1-T3.1-EM1.1-4.2-win	MOKAMAQ-D	none	MOKAMAQ-DM-s	MOKAMAQ-DM-wind	thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
19	MOKAMAQ-DM-V1-T3.1-EM1.1-4.2-win	MOKAMAQ-DM-windingfactor-R0		MOKAMAQ-DM-s	MOKAMAQ-DM-wind	thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
20	MOKAMAQ-DM-V1-T3.1-EM1.1-4.2-win	MOKAMAQ-DM-windingfactor-R0		MOKAMAQ-DM-s	MOKAMAQ-DM-wind	thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
21	MOKAMAQ-DM-V1-T3.1-EM1.1-3.3-StatorDimension-A		MOKAMAQ-DM-slo	MOKAMAQ-DM-stator-E003		thesis-Gaizka Almar	CR	########	v0.1	In Progress	MOKAMA
22	MOKAMAQ-DM-V1-T3.1-EM1.1-2-initialsizing-A006			MOKAMAQ-DM-S	MOKAMAQ-DM-wind	multiphysics simulati	CR	########	v0.1	In Progress	MOKAMA

Fig. 3-8. Knowledge book - MOKA capture phase.

3.4.4 Formalize

This section presents the knowledge representation regarding the product model and process model. The knowledge representation of MOKA has been extended and adapted to ensure fitness for the electric machine development process. The derived profile name is KBeMotorProfile.

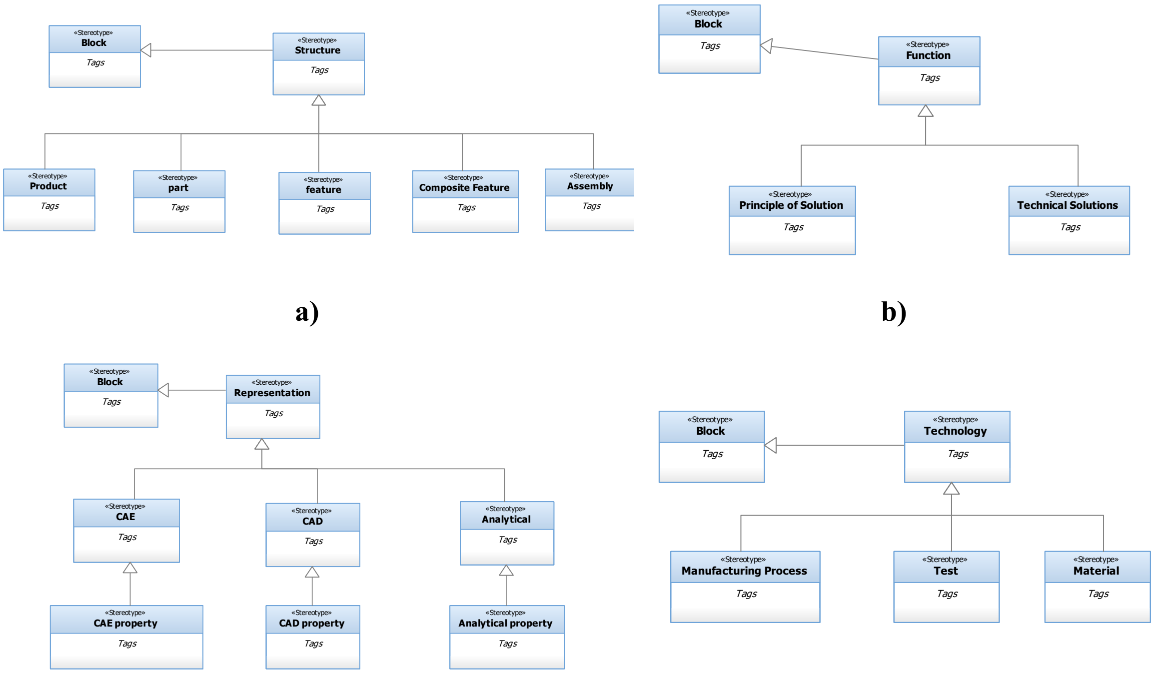
MML (MOKA Modeling Language) is an extension of UML. In this thesis, SysML was selected as the modeling tool instead of UML because of the following reasons. One reason is that SysML makes possible the knowledge representation for the macro-process level. SysML is the standard modeling language of MBSE (see Section 2.2.2). MBSE has become the methodology follow in systems engineering and broadly adopted in academia and industry. Therefore, it is likely a company is already implementing SysML models in their processes making possible the reuse of existing models. Another reason is due to its flexibility of extension through stereotypes and MBSE tools offer integration capabilities with other software tools (Matlab, CADs, and others). Furthermore, the accessibility of many SysML open source and commercial software tools makes it technically feasible to implement it in any project. Current MBSE tools are even capable of generating code automatically. In addition, SysML compared to UML introduces the requirement as a model element having traceability, satisfaction and derivations expanding its use for different systems.

No confusion should arise about modeling a system and modeling knowledge, the KB is independent of the system modeling but it can have models from the system that represent macro-level knowledge. For example, the diagrams shown in Sections 2.3 and 2.4 are considered part of the knowledge base in this work. They represent a system in a high-level form. Nonetheless, their adaptation to a particular system (knowledge implementation) may not require being part of the KB. This is an agreement that should be decided among the actors involved.

The knowledge representation is still divided in the product model and the process model as MOKA's structure. The formalization of both is described in the following subsections.

3.4.4.1 eMachine product model.

The product metamodel shown in Fig. 3-4 divides its views in structure, technology, behavior, function and representation. Fig. 3-9 shows some of the stereotypes added to develop the MML for the product model as a profile extension of SysML. In addition, the association relationships among the objects like has part, is part of, has material, has manufacturing process, etc. are added. Please check appendix C for a complete list of the stereotypes.



a) **b)**

c) **d)**

Fig. 3-9. MML stereotypes as a profile extension of SysML. a) Function; b) Representation c) Structure; d) Technology

The product model is as proposed in MML with the difference of being an extension of SysML rather than UML. New stereotypes are created some are shown in Fig. 3-9. For instance, Test for the technology view, the representation view has CAD, CAE and Analytical as new stereotypes. One additional aspect is the constraint or constraint block. Constraint block is a class of SysML. MML suggests different type of constraints: attribute bound, class bound, association bound, and existence. Instead of following MML guidelines for these cases, Fig. 3-10 shows the new way of representation through associations (class bound) and dependencies for each type of constraint classification. Same semantics but using SysML.

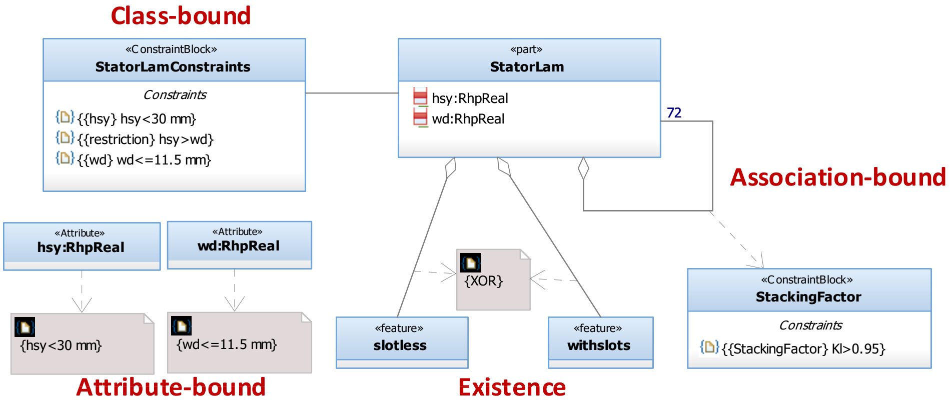


Fig. 3-10. Knowledge representation of constraints for the product model.

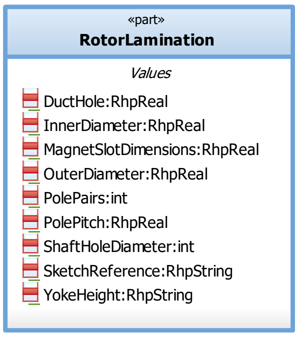
Next, examples that can serve as reference to the reader of the product model with respect to electric machines are given. The behavior view of the product model is intentional omitted since this knowledge representation could be replaced with state machines of SysML. The reader is encourage to read the behavior view in MOKA.

Structure view:

This section summarizes the structure view by showing some relevant diagrams and relationships about the electric machine product model. This view describes the decomposition of the motor into assembly, parts and design features. The highest level of decomposition is the product, which represents the entire electric machine, next, different machine parts or subassemblies constitute the assembly, and a part is the lowest level of decomposition. Composite feature and features represent design features of a part.

Fig. 3-13 shows the eMachine structure classes such as parts (e.g. stator and rotor lamination, permanent magnet, bearing), assemblies (e.g. stator core, shaft), composite features and features (e.g. slots, magnet slots) and their relationships among other objects. In this case, to limit the knowledge base and to avoid introducing unnecessary information, devices that their decomposition was not required are considered parts e.g. sensors and actuators, fans etc. Of course, this decomposition may change to the convenience of the knowledge engineer.

The features shown in Fig. 3-13 can also be represented as attributes. The main reason to create a feature is only to leverage its use as an object. However, if the need of an object is not required it can be used as an attribute as illustrated for the rotor lamination in Fig. 3-11.



«part»
RotorLamination

Values

DuctHole:RhpReal
InnerDiameter:RhpReal
MagnetSlotDimensions:RhpReal
OuterDiameter:RhpReal
PolePairs:int
PolePitch:RhpReal
ShaftHoleDiameter:int
SketchReference:RhpString
YokeHeight:RhpString

Fig. 3-11. Rotor lamination object and attributes.

Function view:

The function view can become the most extensive, since a set of functions can be described for each part (or assembly). This consist of a break down of functions with their principle of solution and technical solutions, in this way the user can be guided to select a specific solution fulfilling his requirements. In addition, some general constraints are shown in this view although these can be included in any view as long as they help to communicate what is needed. To show how the knowledge is represented let us concentrate on the rotor structure as seen in Fig. 3-13. The diagrams are shown from bottom to top i.e. starting from the rotor lamination, then the rotor core and so on up to the whole rotor assembly.

Fig. 3-14 shows the functions for the rotor lamination. One of the functions the rotor lamination has is to define the rotor core pattern. This pattern refers to the cross-section of the rotor that depends on the dimensions, number of poles, and typology (non-salient or interior permanent magnet). The selection of the number of assembled sheets defines the rotor core length. Another function is to enhance the magnetic flux flow. These functions are realized through the sizing process of the motor, where the designer typically solves out using analytic models or FEM analysis.

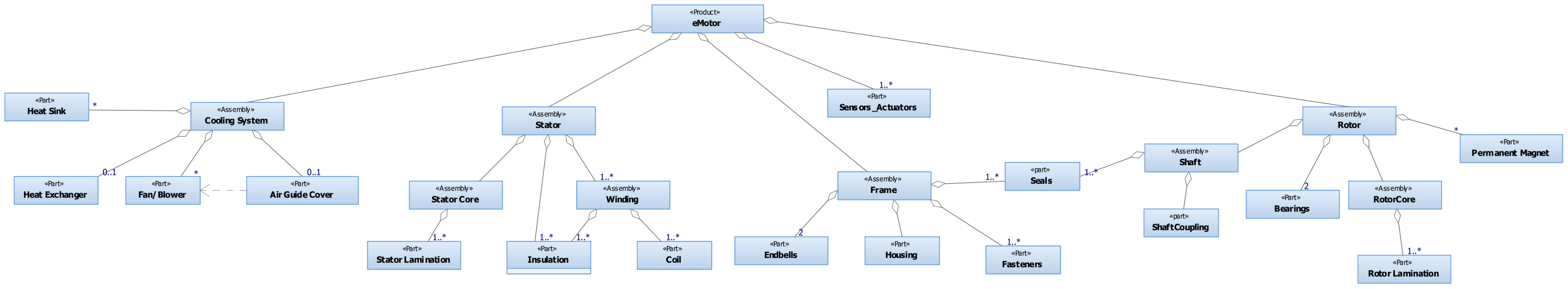
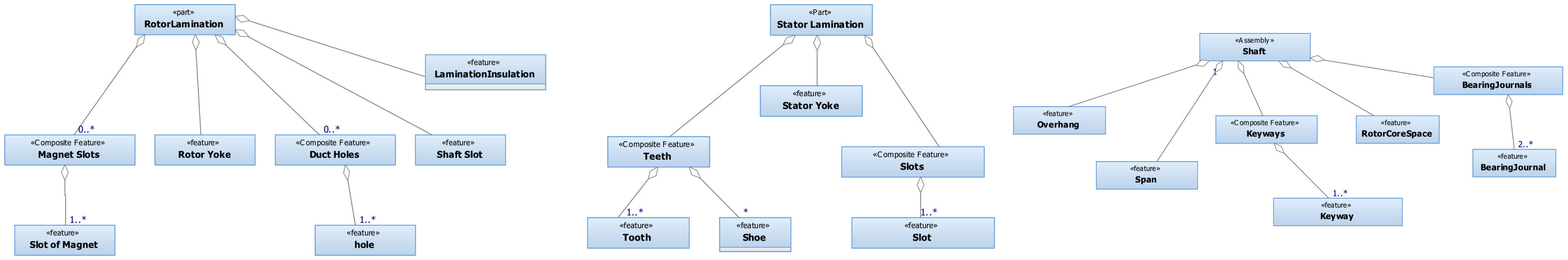


Fig. 3-12. eMotor structure model.



a) b) c)

Fig. 3-13. a) Rotor lamination structure; b) Stator lamination structure; c) Shaft structure

On the other hand, in order to enhance the magnetic flux flow the designer as a principle of solution selects one among different types of soft magnetic materials where the selected material is a technical solution to the function

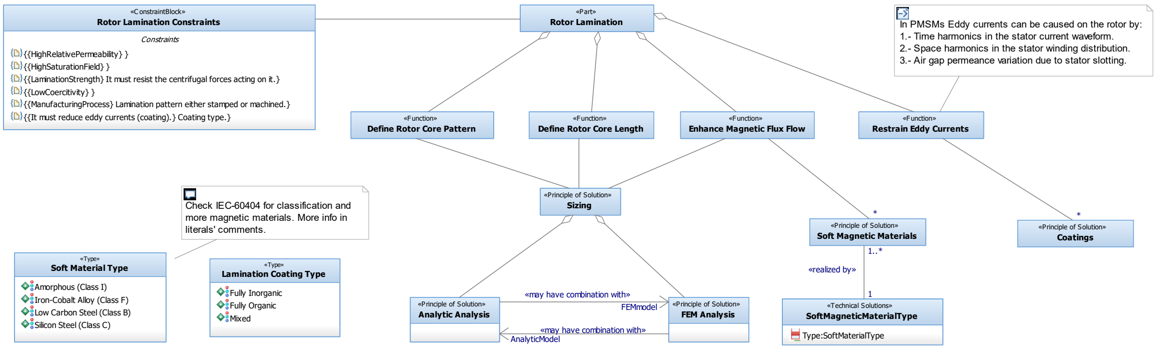


Fig. 3-14. Rotor lamination function view example.

The last function the rotor lamination shall satisfy is to restrain eddy currents, this is done by insulation coatings on the lamination surface, which in this case is realized by the selection of one of the different coatings type. Now, the rotor core (see Fig. 3-15) shall retain the magnet poles. To make this possible, mechanical retaining method or chemical retaining methods or a combination of both can be used.

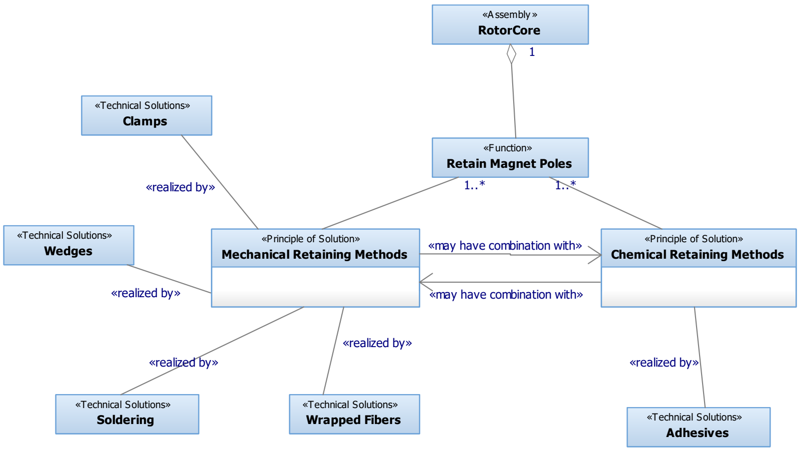


Fig. 3-15. Rotor core function view example.

Following the same process the rest of the rotor is shown in Fig. 3-16 and Fig. 3-17. Apart from the visible information in all the aforementioned figures, within the objects it is possible to have descriptions of steps, rationales or links to source documents of any part, Fig. 3-17d shows a description example.

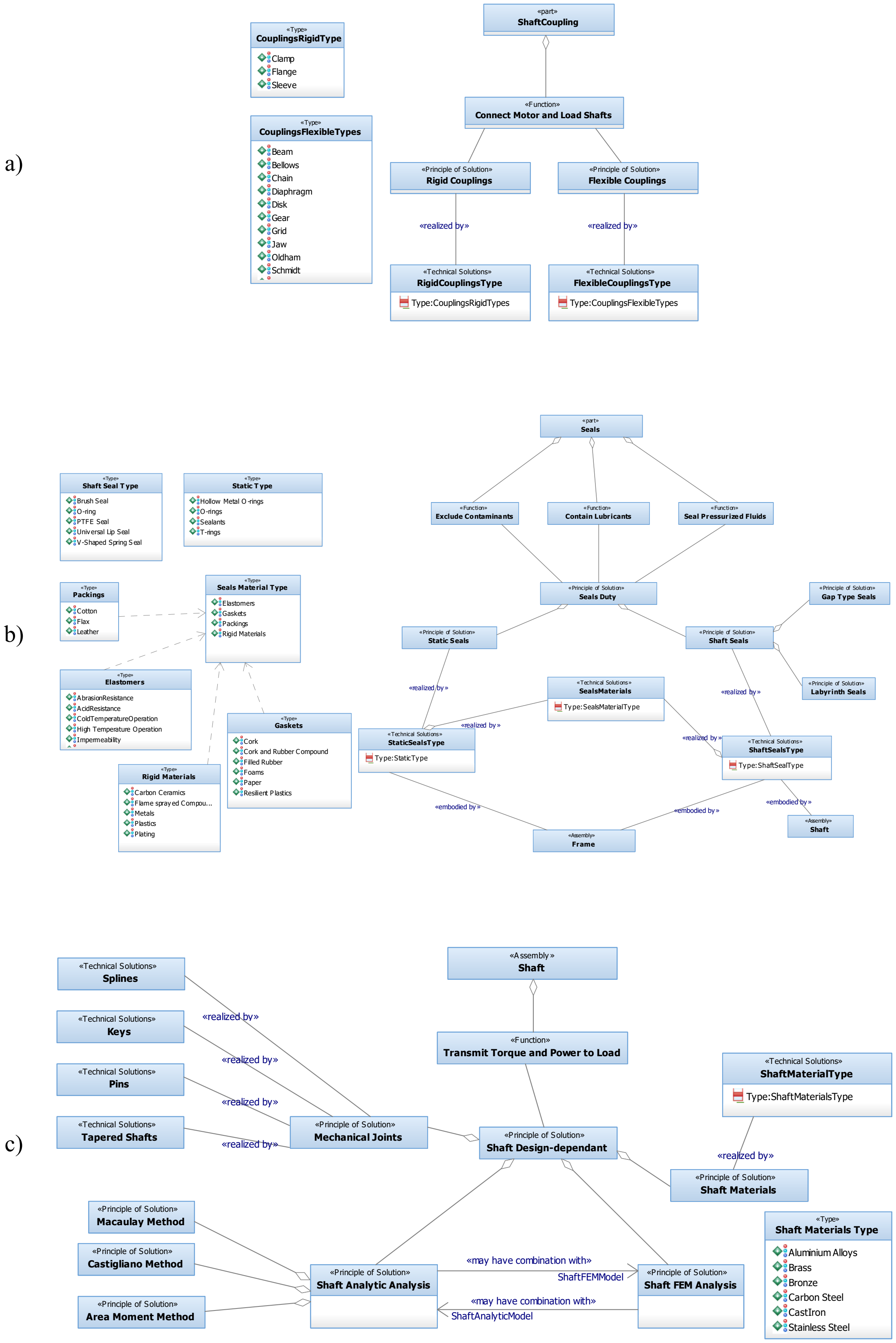
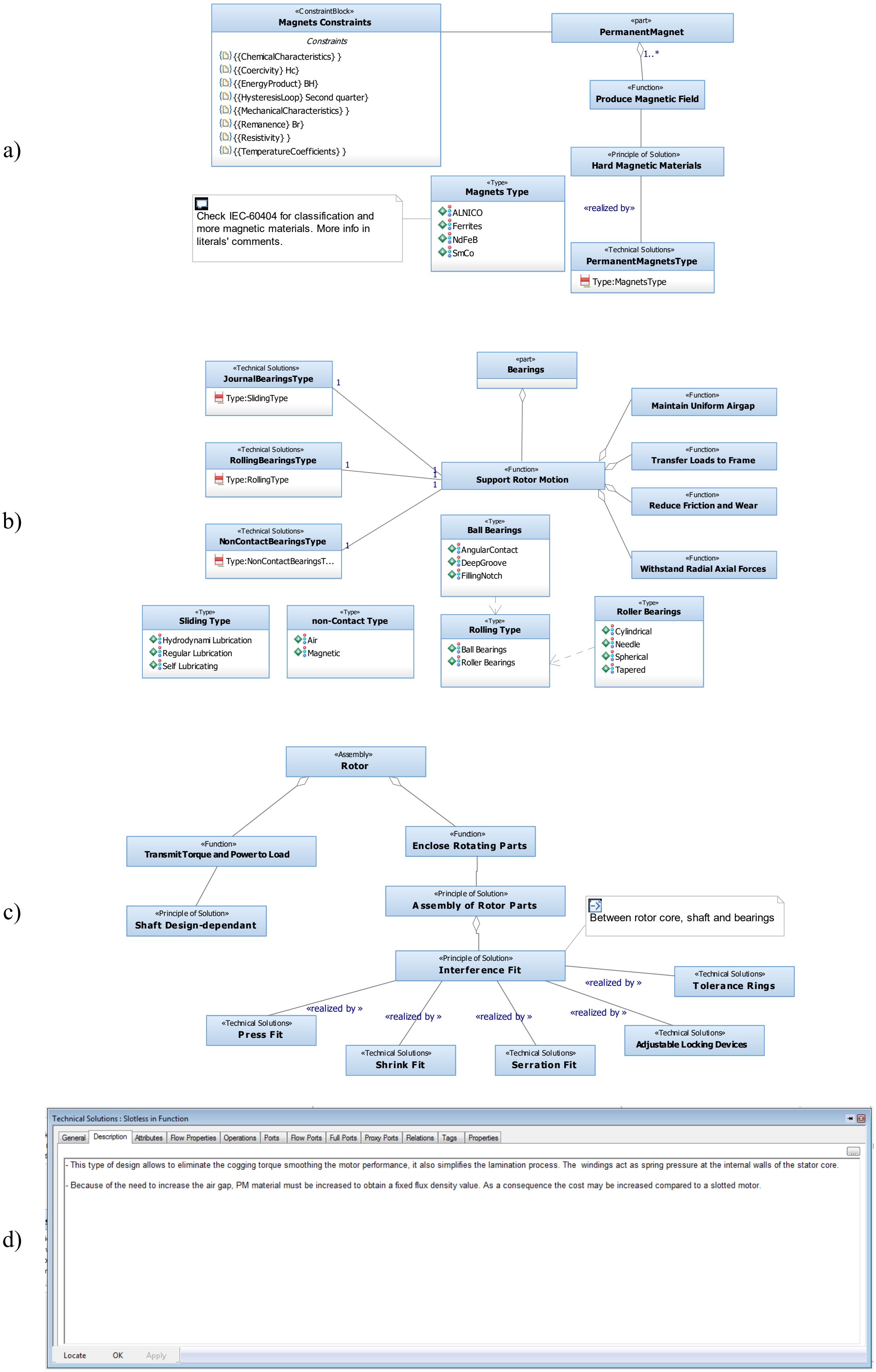


Fig. 3-16. Function view for a) shaft couplings, b) Seals, c) shaft.



- This type of design allows to eliminate the cogging torque smoothing the motor performance, it also simplifies the lamination process. The windings act as spring pressure at the internal walls of the stator core.

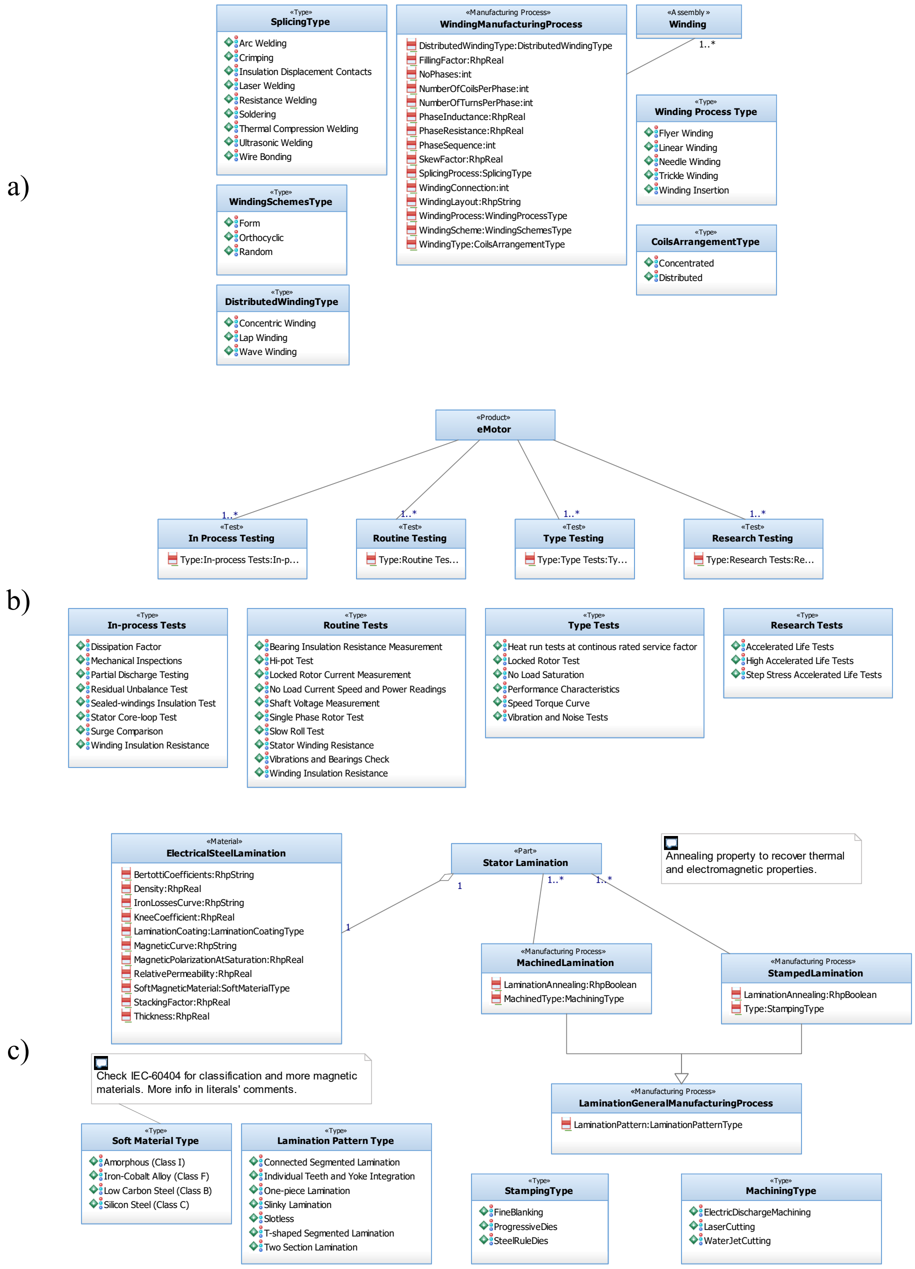
- Because of the need to increase the air gap, PM material must be increased to obtain a fixed flux density value. As a consequence the cost may be increased compared to a slotted motor.

Fig. 3-17. Function views for a) magnets, b) bearings, c) rotor, d) object description.

Technology view:

This view has the classes of manufacturing process, material and technology. The test class is added as an extension; see Fig. 3-9d. This class intends to cover any type of test during the performance testing of a motor e.g. thermal test, load test, no load test. Fig. 3-18b shows typical test carry out on electric motors.

Moreover, the function view presented technical solutions that are closely related to manufacturing process and materials. These data types are shared between both views. Fig. 3-9c illustrates one example with the soft magnetic materials type.



Annealing property to recover thermal and electromagnetic properties.

Check IEC-60404 for classification and more magnetic materials. More info in literals' comments.

Fig. 3-18. Technology view: a) winding, b) tests, c) stator lamination

Representation view:

The representation view depends on each case of application. It means that depending on the KBE application that wants to be developed some parameters of an entity may be discarded in the representation view. The extended stereotypes of this view has been set as CAD, CAE and Analytical as seen in Fig. 3-9c. This is devised in this way since the electric machine is modeled using these types of tools. The CAD stereotype is used to describe any type of geometry and CAD files. CAE is used for any model represented in FEM, CFD or any other tool that suits this category. Finally, the analytical stereotype regards the creation of particular functions or models developed outside any of the other categories e.g. Simulink models, VB, python, Matlab. Fig. 3-19 shows the general attributes inherited by CAE models for FEM_FLUXGKB and LumpedCktTool_MOTORCAD. Fig. 3-20 shows the properties related for the CAE tool FEM_FLUXGKB.

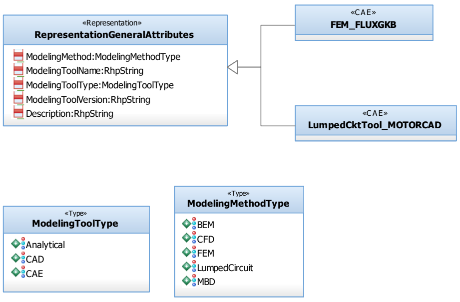


Fig. 3-19. Inheritance example of general attributes.

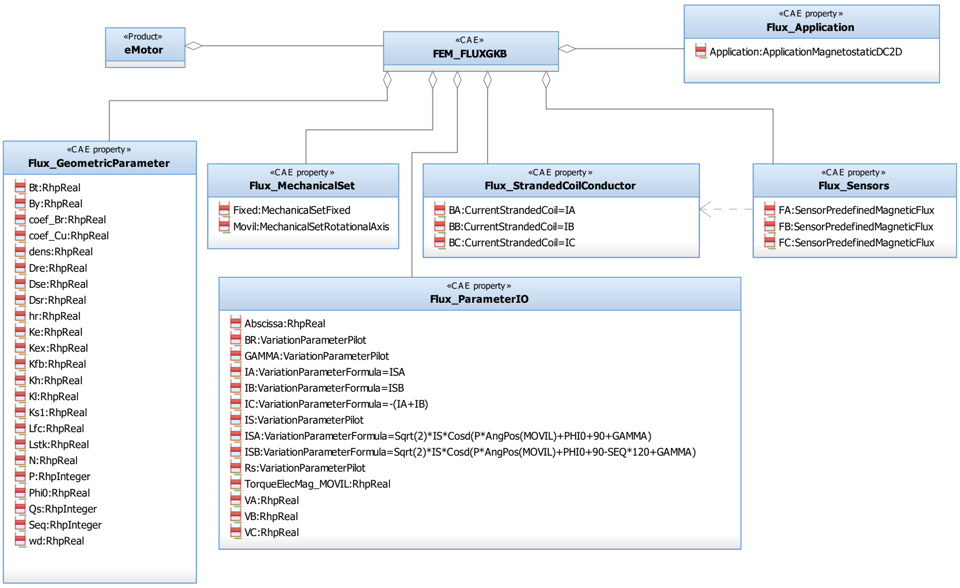
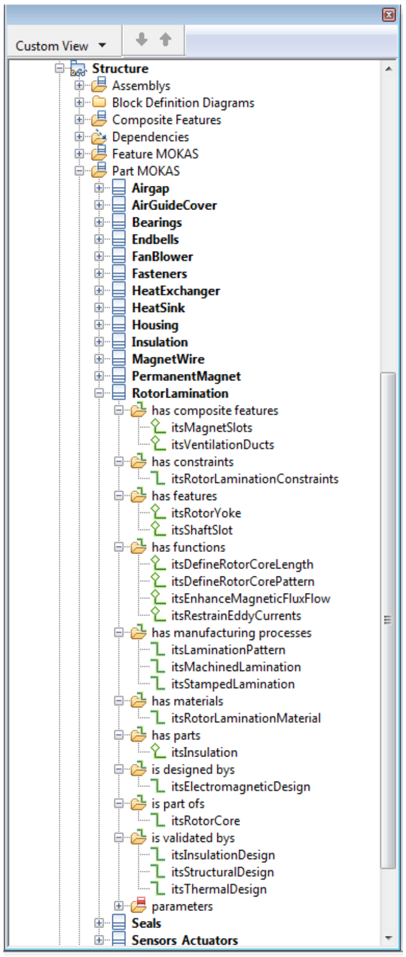


Fig. 3-20. Representation view of FEM tool Flux.

Finally, the associations among the object are visualized with more clarity through the tree view of the browser the tool provides. Fig. 3-21 shows the tree view of the rotor lamination where these associations can be read.



- Structure
  - Assemblys
  - Block Definition Diagrams
  - Composite Features
  - Dependencies
  - Feature MOKAS
  - Part MOKAS
    - **Airgap**
    - **AirGuideCover**
    - **Bearings**
    - **Endbells**
    - **FanBlower**
    - **Fasteners**
    - **HeatExchanger**
    - **HeatSink**
    - **Housing**
    - **Insulation**
    - **MagnetWire**
    - **PermanentMagnet**
    - **RotorLamination**
      - has composite features
        - itsMagnetSlots
        - itsVentilationDucts
      - has constraints
        - itsRotorLaminationConstraints
      - has features
        - itsRotorYoke
        - itsShaftSlot
      - has functions
        - itsDefineRotorCoreLength
        - itsDefineRotorCorePattern
        - itsEnhanceMagneticFluxFlow
        - itsRestrainEddyCurrents
      - has manufacturing processes
        - itsLaminationPattern
        - itsMachinedLamination
        - itsStampedLamination
      - has materials
        - itsRotorLaminationMaterial
      - has parts
        - itsInsulation
      - is designed bys
        - itsElectromagneticDesign
      - is part ofs
        - itsRotorCore
      - is validated bys
        - itsInsulationDesign
        - itsStructuralDesign
        - itsThermalDesign
      - parameters
    - **Seals**
    - **Sensors Actuators**

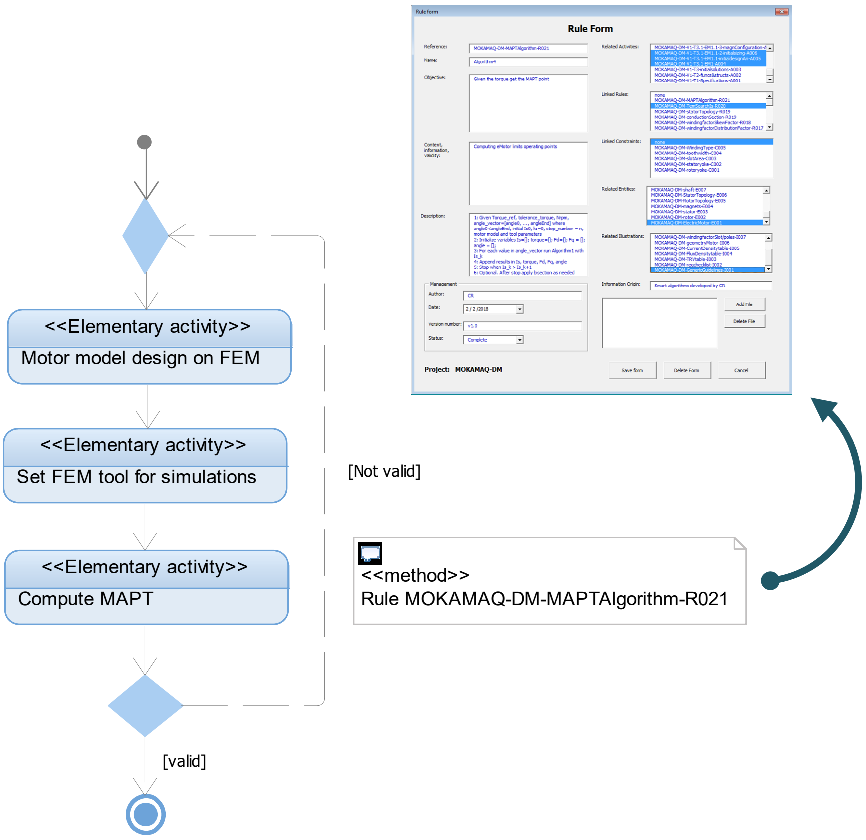
Fig. 3-21. Tree view of rotor lamination in Rhapsody.

3.4.4.2 eMachine process model

MOKA suggests the use of activity diagrams to accomplish the knowledge formalization of processes and extends UML activity stereotypes with compound activity and elementary activity. The former to represent nested actions belonging to a higher activity and the latter to represent a single action. In addition, it suggests the following attributes for each activity.

1. Description: rationale of an activity.
2. Inputs and outputs: reference to instance objects.
3. Locals: process data not contained in the product model.
4. Constraints: soft (open to modifications), hard (mandatory) and user constraints (any other)
5. Method: rules associated with an activity.

One example can be seen in Fig. 3-22 with a method beside an elementary activity that links a method to a rule form of the knowledge book.



Rule Form

Reference: MOKAMAQ-DM-MAPTAlgorithm-R021

Name: Algorithm4

Objective: Given the torque get the MAPT point

Context, information, validity: Computing eMotor limits operating points

Description:
1: Given Torque_ref, tolerance_torque, Nrpm, angle_vector=[angle0, ..., angleEnd] where angle0<angleEnd, initial Is0, k:=0, step_number = n, motor model and tool parameters
2: Initialize variables Is=[]; torque=[]; Fd=[]; Fq = []; angle = [];
3: For each value in angle_vector run Algorithm1 with Is_k
4: Append results in Is, torque, Fd, Fq, angle
5: Stop when Is_k > Is_k+1
6: Optional. After stop apply bisection as needed

Management
Author: CR
Date: 2 / 2 /2018
Version number: v1.0
Status: Complete

Related Activities:
MOKAMAQ-DM-V1-T3.1-EM1.1-3-magnConfiguration-A
MOKAMAQ-DM-V1-T3.1-EM1.1-2-initialsizing-A006
MOKAMAQ-DM-V1-T3.1-EM1.1-initialdesignAn-A005
MOKAMAQ-DM-V1-T3.1-EM1-A004
MOKAMAQ-DM-V1-T3-initialsolutions-A003
MOKAMAQ-DM-V1-T2-funcs&structs-A002
MOKAMAQ-DM-V1-T1-Specifications-A001

Linked Rules:
none
MOKAMAQ-DM-MAPTAlgorithm-R021
MOKAMAQ-DM-TemSearchIs-R020
MOKAMAQ-DM-statorTopology-R019
MOKAMAQ-DM-conductionSection-R019
MOKAMAQ-DM-windingfactorSkewFactor-R018
MOKAMAQ-DM-windingfactorDistributionFactor-R017

Linked Constraints:
none
MOKAMAQ-DM-WindingType-C005
MOKAMAQ-DM-toothwidth-C004
MOKAMAQ-DM-slotArea-C003
MOKAMAQ-DM-statoryoke-C002
MOKAMAQ-DM-rotoryoke-C001

Related Entities:
MOKAMAQ-DM-shaft-E007
MOKAMAQ-DM-StatorTopology-E006
MOKAMAQ-DM-RotorTopology-E005
MOKAMAQ-DM-magnets-E004
MOKAMAQ-DM-stator-E003
MOKAMAQ-DM-rotor-E002
MOKAMAQ-DM-ElectricMotor-E001

Related Illustrations:
MOKAMAQ-DM-windingfactorSlot/poles-I007
MOKAMAQ-DM-geometryMotor-I006
MOKAMAQ-DM-CurrentDensitytable-I005
MOKAMAQ-DM-FluxDensitytable-I004
MOKAMAQ-DM-TRVtable-I003
MOKAMAQ-DM-regchecklist-I002
MOKAMAQ-DM-GenericGuidelines-I001

Information Origin: Smart algorithms developed by CR

Project: MOKAMAQ-DM

Fig. 3-22. Activity linked to rule form outside the knowledge modeling tool.

According to MOKA the design process can be approached at the macro level and micro level view. However, Stokes et. al in [28] page 220 state that "MML relates more specifically to the micro view, providing sufficient detail to enable them to be automated". Besides MML relates specifically to the micro level view, its representation has limitations such as linking to the knowledge book rather than having the knowledge at hand on the same modeling tool. This makes difficult the knowledge retrieval from one software. Instead, you may need to jump among software (or paper) in order to understand the problem. MOKA only offers guideline for the design process model not a metamodel as with the product model. It suggests the use of UML activity diagrams for process representation and adds the classes of elementary and compound activity. SysML activity diagrams offer more options with actions call behavior and call operations. The eMachine process model is divided in the macro-process and the micro-process level. The knowledge representation for the eMachine process model is now described.

Macro-process level. As shown in Fig. 3-23, the macro-process knowledge representation encloses the SysML profile and the KBeMotorProfile. The KBeMotorProfile intends to complement the macro level view by modeling knowledge for the micro level view and it is explain later. SysML is the selected modeling language for the macro level view. The motivation for its selection were previously described.

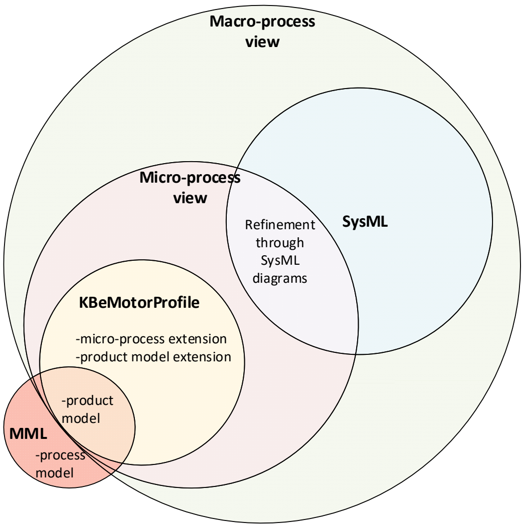


Fig. 3-23. Macro and micro process profiles relations

The macro-process view for the eMachine development process is conceived to cope with knowledge modeling regarding KBE applications integration with other entities (SW, end-users, etc.) to pursue a goal. This overcomes the shortcoming of MOKA of not showing the usage of KBE applications in the design process. Furthermore, knowledge regarding system design can also be modeled to make it part of the generic KB. For example, the requirement-engineering framework presented in Chapter 2 is a result of a thorough knowledge structure modeling from the captured knowledge regarding the approach the consulted experts carry out to undertake this stage of the development process. The proposed requirement-engineering framework is an example of macro-process knowledge representation. This requirement engineering framework is part of the KB.

To make a macro-process part of the generic KB. In this thesis, it is suggested considering knowledge modeling for macro-processes with static knowledge or with controlled and predictable changes or transform the knowledge into a higher-level form. Knowledge that changes dynamically cannot be part of the generic KB. For instance, assume that your company has two development teams of electric motors. Team A develops motors for elevation systems and team B for electric vehicles. Both use their own requirement framework. However, team A develops in the industrial V-cycle having always the same requirements fields (not values) and team B has contracts only for the concept V-cycle, thus their requirement fields changes over time (for instance cooling system type, materials, geometries). Then, the requirement framework of team A is worth of consideration for its incorporation to the generic KB. On the other hand, not the team's B framework. Another option is to capture the knowledge of both and make a higher-level requirement framework achieving a framework with a common ground. Like the one presented in Chapter 3. Chapter 6 gives an implementation example for the macro-process level.

Micro-process level. The KBeMotorProfile is devised to overcome MOKA's shortcoming about the micro-process model. The Venn diagram in Fig. 3-23 shows that the micro-process view depends on the KBeMotorProfile and can be refine through SysML diagrams such as parametric diagrams, requirements diagrams, sequence diagrams, etc. The KBeMotorProfile contains the MML product model. In addition to the previously mentioned extensions of the product model, a new metamodel is proposed to represent knowledge in the micro-process level. This metamodel is shown in Fig. 3-24. This metamodel is derived after a thorough analysis of the problem solving cycle as a micro-cycle according to VDI-2206 [85] and illustrated in Fig. 3-25a. Supposing that the knowledge for a particular process is well-defined. Therefore, objectives are stablished. The resulting reasoning is the following.

Starting from the premise our objective is defined then to accomplish this objective one or several approaches for a solution can be carried out. Each approach is undertaken through a set of indications with given criteria. One indication can also have criteria or can be composed of more indications (expanded indications). Finally, the following argument should always be considered: An indication can be transformed into an objective if an indication of one objective contains more indications (expanded indication) and it can be understood out of the context of the container objective. Therefore, this indication can stand by itself and it can be transformed into an objective. Examples are given regarding this issue during the implementation phase, presented in Chapter 5.

The main point is the knowledge engineer be consistent throughout the knowledge formalization.

The definition of each concept in the context of the metamodel is now described.

1. Objective: a micro goal, task or plan to achieve a purpose.
2. Approach: a particular manner of addressing the objective. It encloses one or a group of indications.
3. Indication: It is a step, suggestion, hint, rule, or principle of solution for a particular approach.
4. Criteria: It enclose the rationale, conditions, exceptions, description or facts for an approach or indication.
5. Domain: Specific sphere of knowledge. This stereotype was particularly introduced for the six design domains in the electric motor V-cycle. However, the author leaves it open for any knowledge sphere.
6. Cross-domain objective: objective involving two or more domains.

7. Requirement: requirement of the KBE application.

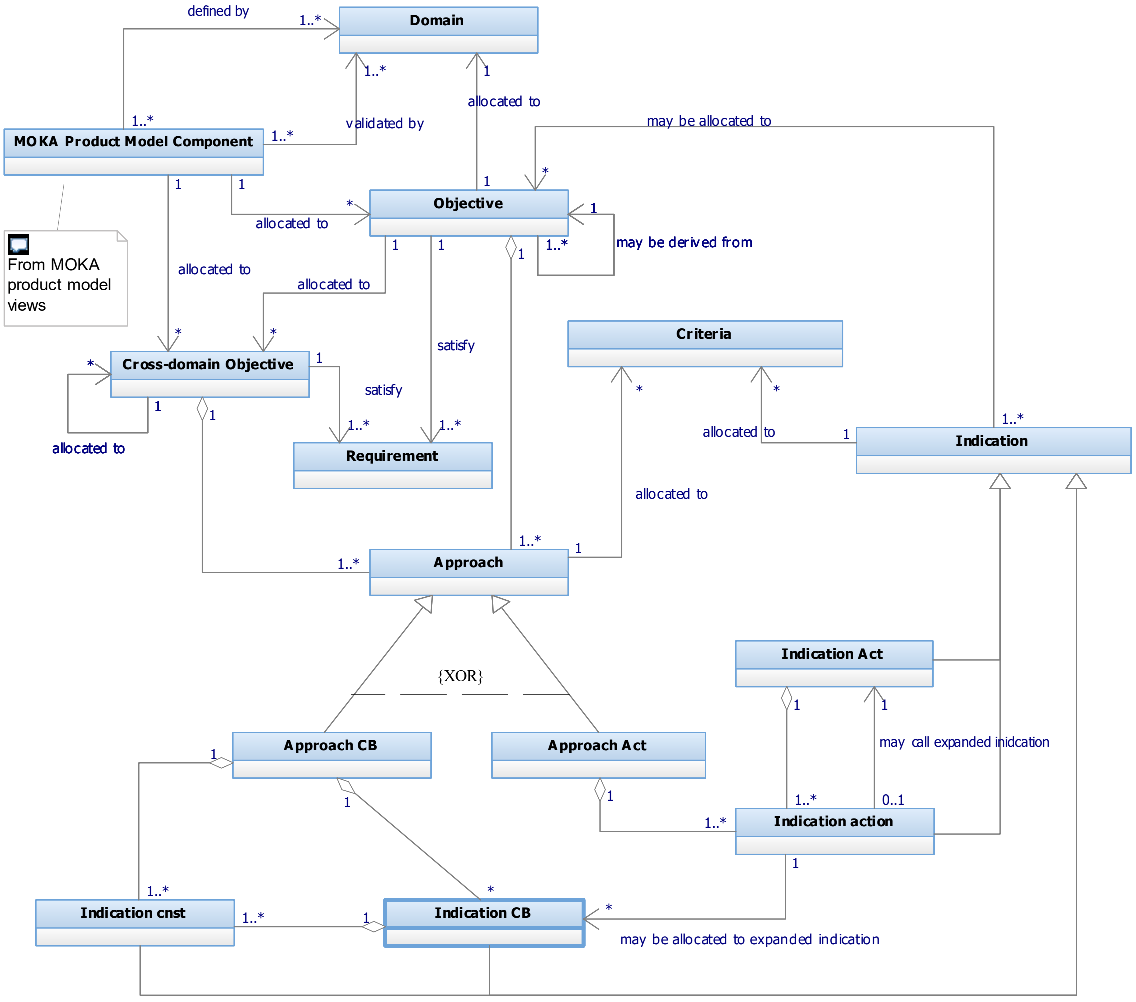
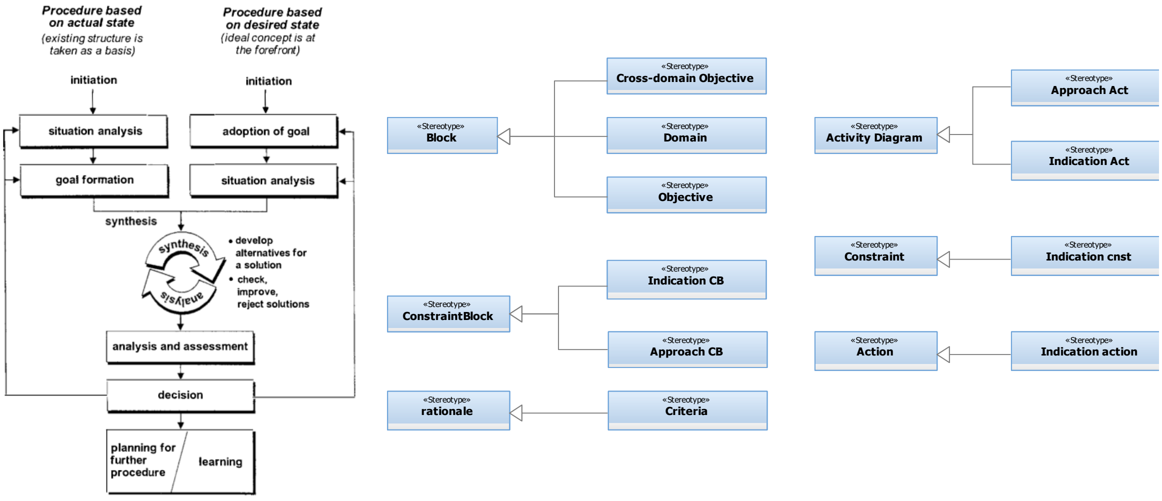


Fig. 3-24. Micro-process metamodel.



a) b)

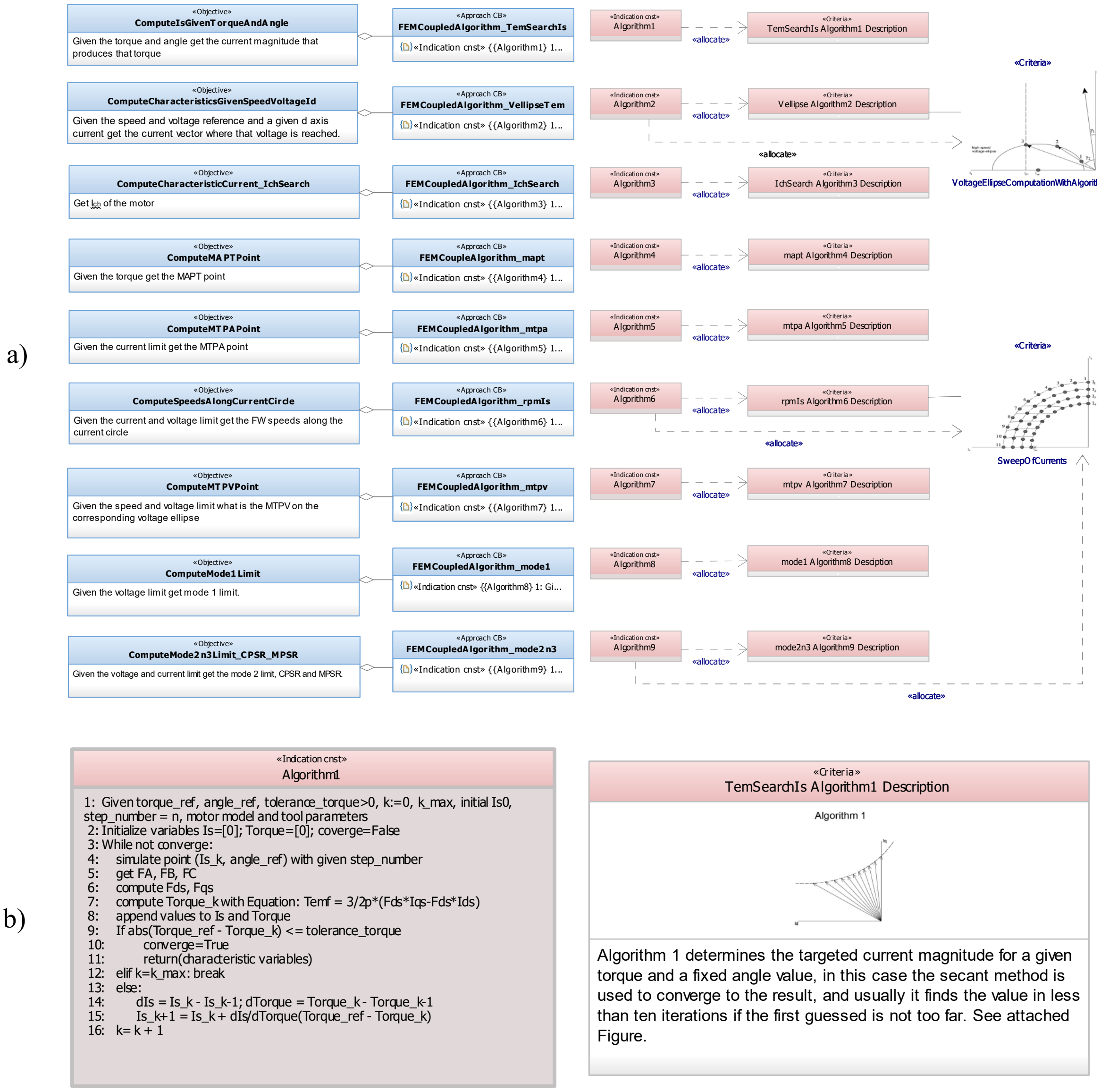
Fig. 3-25. a) problem solving micro cycle VDI-2206 [85]. b) stereotypes for the micro-process metamodel.

Fig. 3-25b presents the stereotypes classification as an extension of SysML for each term. To explain the proposed micro-process metamodel let us start from the objective. The objective is a specialization stereotype of the class block. The objective is allocated to a domain. It can be allocated to none or several cross-domain objectives. An objective satisfies one or more requirements. An objective can be derived from another objective. An objective contains one or more approaches. Continuing with the approach, an approach is specialized in two classes, approach CB and approach Act. For a given approach, one route between the two must be taken. Indication is a generalization of four classes, Indication CB, Indication cnst, Indication Act and Indication action. Their use depends on the approach selection. An indication can also be allocated to the criteria class.

Approach CB is a specialization of the class constraint block. An approach can be allocated to a criteria stereotype. The criteria is a specialization of the SysML rationale. The purpose of Approach CB is to allow the creation of indications in text format suited for those indications relating rules in equation form or algorithms difficult to show through activity diagrams. Besides, by using this type of representation the knowledge engineer can elaborate more among several rules (equations) using parametric diagrams (example in Section 5.3). Approach CB contains Indication cnst of the SysML class constraint and may contain Indication CB of the type constraint block to expand a particular Indication cnst.

To illustrate one of the advantages of this knowledge representation first check Fig. 3-22. This figure follows MOKA guideline for process representation. It can be seen that the activity call “ComputeMAPT” is linked to a rule form saved in the knowledge book (implemented in MS Excel in this thesis). The rule form is put in the figure for illustration purpose; however, the form is outside the knowledge modeling software. Therefore, the user must literally look for that rule in the knowledge book making the knowledge representation less readable in the same modeling tool.

On the other hand, following the proposed representation (for this particular example the Approach CB) the algorithm can be included in the same model. Fig. 3-26a shows this example, the figure shows how each objective contains the purpose of each algorithm, and each objective is linked through aggregation and “has approach” association end (check Appendix C) relationship to the Approach CB which contains an indication cnst type with the algorithm. Fig. 3-26b shows an example of an algorithm illustration within an Indication cnst, at the same time this Indication cnst can be allocated to Criteria. Fig. 3-26b shows an example of Criteria belonging to algorithm 1. Then, the activity mentioned of Fig. 3-22 can be allocated to its respective objective having all the knowledge representation at hand in the same modeling tool. This section also provides examples of this type of allocation but first the Approach Act must be explained.



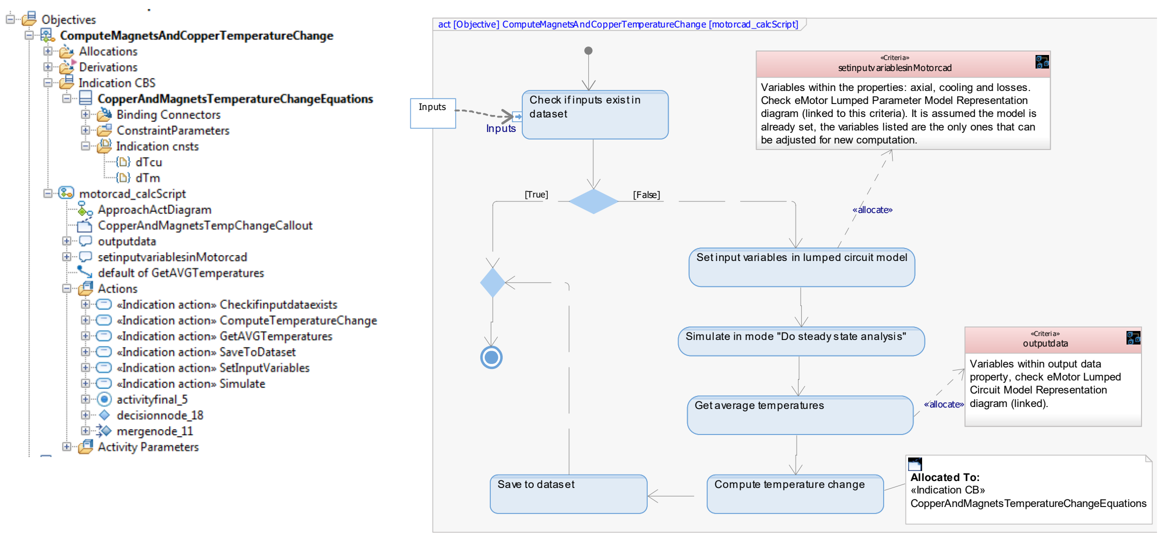
```
Given torque_ref, angle_ref, tolerance_torque>0, k:=0, k_max, initial Is0,
step_number = n, motor model and tool parameters
Initialize variables Is=[0]; Torque=[0]; coverge=False
While not converge:
    simulate point (Is_k, angle_ref) with given step_number
    get FA, FB, FC
    compute Fds, Fqs
    compute Torque_k with Equation: Temf = 3/2p*(Fds*Iqs-Fds*Ids)
    append values to Is and Torque
    If abs(Torque_ref - Torque_k) <= tolerance_torque
        converge=True
        return(characteristic variables)
    elif k=k_max: break
    else:
        dIs = Is_k - Is_k-1; dTorque = Torque_k - Torque_k-1
        Is_k+1 = Is_k + dIs/dTorque(Torque_ref - Torque_k)
    k= k + 1
```

Algorithm 1 determines the targeted current magnitude for a given torque and a fixed angle value, in this case the secant method is used to converge to the result, and usually it finds the value in less than ten iterations if the first guessed is not too far. See attached Figure.

Fig. 3-26. a) Knowledge representation example of the approach CB route. b) Expanded indication cnst and criteria.

Following the proposed metamodel shown in Fig. 3-24. The Approach Act is a specialization of the class activity diagram. It contains Indication actions, a specialization of the class action (see Fig. 3-25b). For this case, the expansion of an indication can be done through either the Indication Act or the Indication CB. Indication Act is a specialization of activity diagram and Indication CB is a specialization of constraint block. The Indication Act is referenced to the container indication through a call behavior action and for the Indication CB this is done through allocation. Fig. 3-27 shows an example for the Approach Act. For this case, the browser view of the modeling tool (Fig. 3-27a) is added in order to view the different classes since the indication actions stereotype cannot be visualized directly on the diagram.

The diagram is now summarized. The objective "compute magnets and copper temperature change" has the solution approach name "motorcad_calcScript". This approach has the six indication actions shown within the diagram, where two of these indications have criteria. In addition, the indication “Compute temperature change” is extended through the Indication CB with name “Copper and magnets temperature change equations” containing two <<indications cnst>>. In the same form, an indication can be allocated to an objective if this indication can stands by itself as an objective.



Variables within the properties: axial, cooling and losses. Check eMotor Lumped Parameter Model Representation diagram (linked to this criteria). It is assumed the model is already set, the variables listed are the only ones that can be adjusted for new computation.

Variables within output data property, check eMotor Lumped Circuit Model Representation diagram (linked).

a) b)

Fig. 3-27. a) Browser screenshot (objective with approach Act). b) Knowledge representation example with approach act.

Fig. 3-28 presents examples of indications allocated to objectives. The first indication action in the approach act diagram is allocated to the objective previously mentioned in Fig. 3-27. The objective to which is allocated can be used for different contexts and the process of computation will not change.

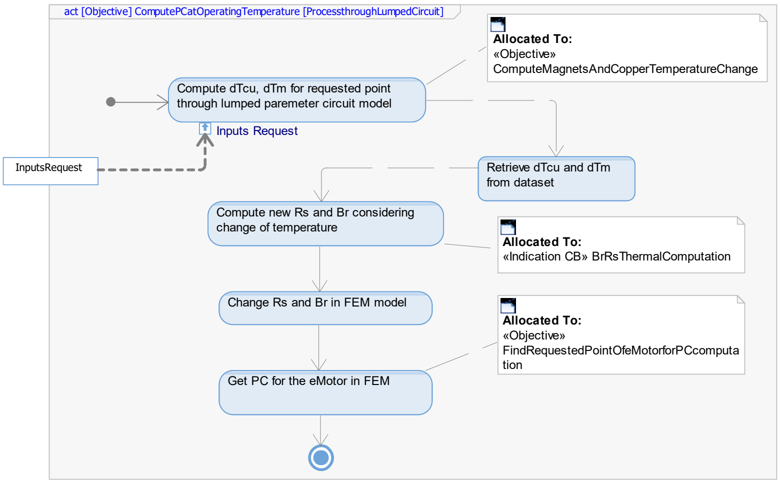
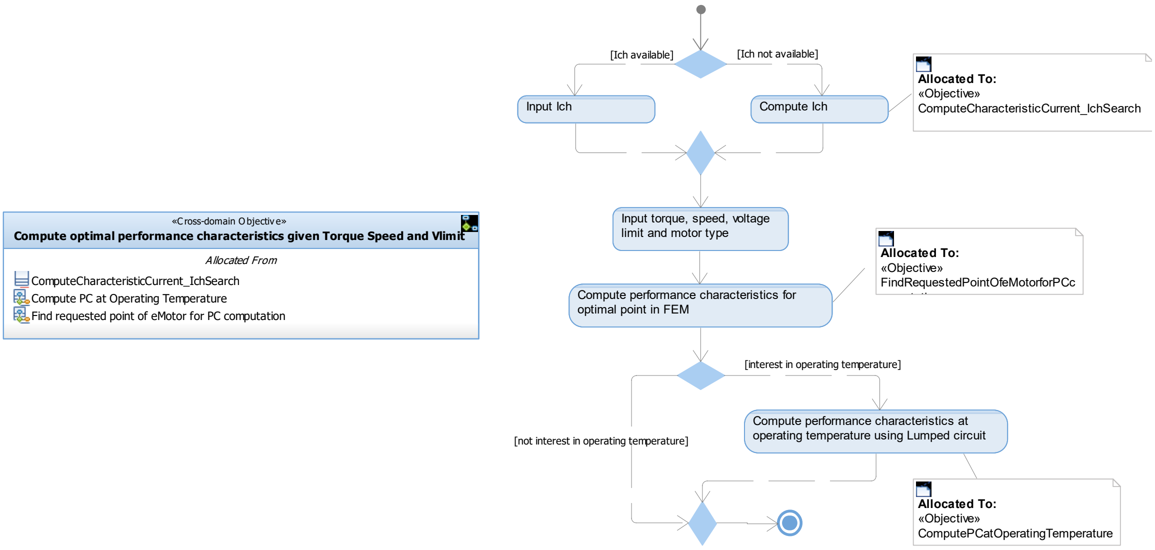


Fig. 3-28. Knowledge representation including indications allocated to objective.

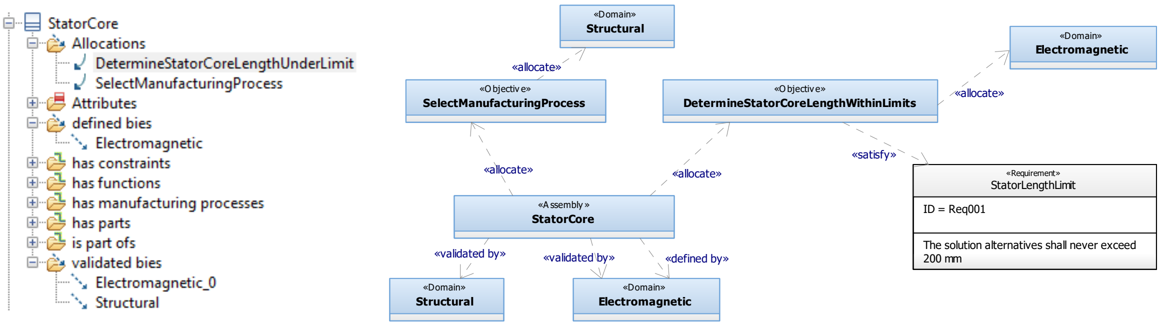
The cross-domain objective follows the same structure of an objective. However, this is bounded by the objectives and cross-objectives allocated to it. Fig. 3-29 illustrates one example where the electromagnetic and thermal domains are involved in the process of computation of performance characteristics of an eMotor. Fig. 3-29a shows the objectives allocated to the cross-domain objective and Fig. 3-29b shows the Approach act with name “electromagnetic and thermal computation” in which these objectives are applied.



a) b)

Fig. 3-29. Cross-domain objective example with approach act. a) Allocated objectives. b) Approach act.

Finally, the MOKA product model component shown in the metamodel in Fig. 3-24 is a generalization of the product model objects. The dependencies <<defined by>> and <<validated by>> to specific domains provide a way to link the product model and the process model for the cases the objects from the product model are involved in a process. Besides, the component is allocated to objectives or cross-domain objectives to define the how to undertake the desired process involving the component by the specified domains. Fig. 3-30 gives an example of these associations where the stator core is defined by the electromagnetic domain and validated by the structural and electromagnetic domains. The tree view of the browser in Fig. 3-30a shows these relationships with more clarity. The stator core is allocated to the objectives “select manufacturing process” and “determine stator core length within limits”. The objectives are allocated to one specific domain. The same figure illustrates the objective “determine stator core length within limits” that satisfies the requirement “stator length limit”.

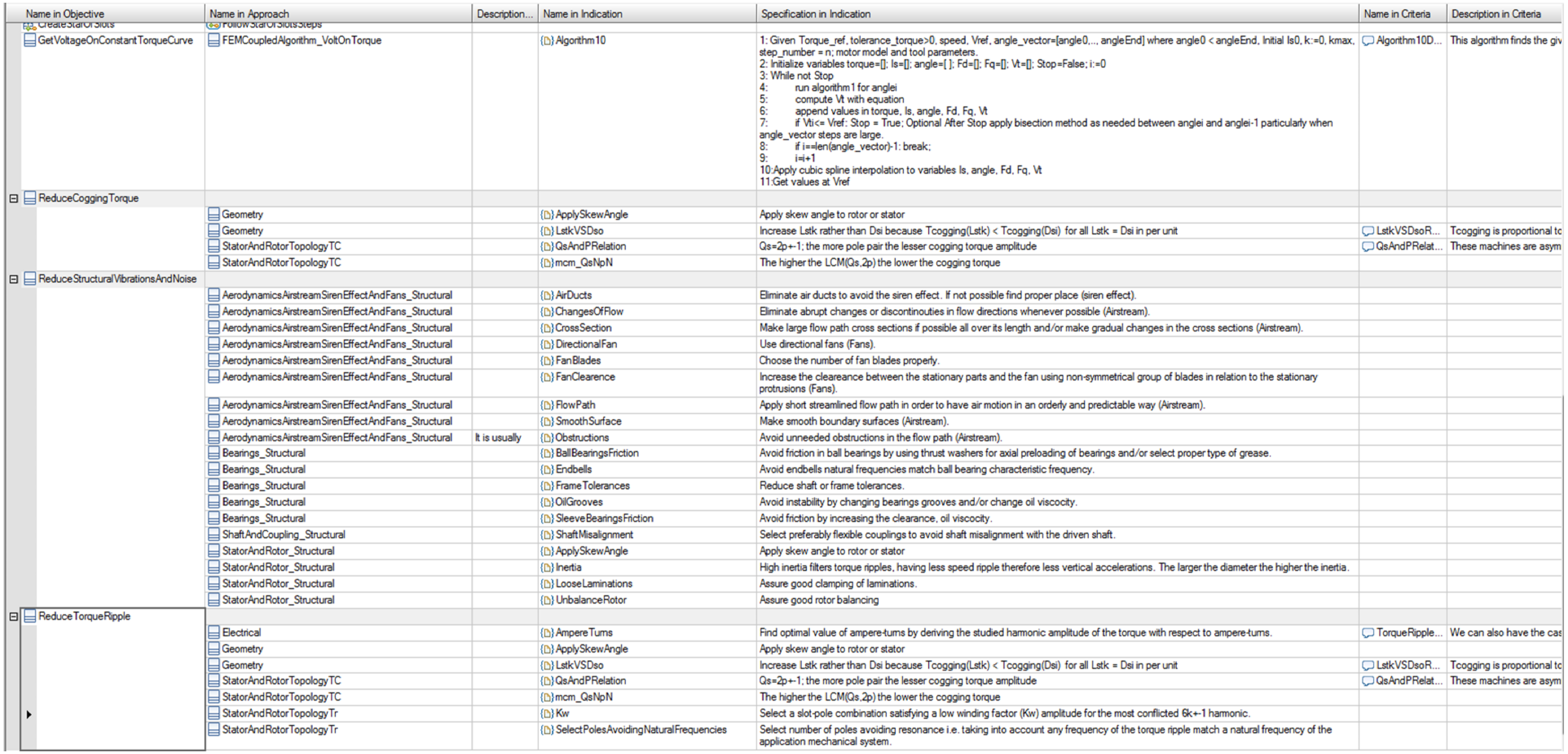


a) b)

Fig. 3-30. a) Browser screenshot. b) Objective relationships with requirements, components and domains.

The proposed micro-process metamodel not only gives the capability to represent knowledge in an intuitive manner but also provides the advantage of retrieving knowledge through relational tables that can be read and understood by any actor. IBM Rhapsody, the tool herein used, has the capability of using context pattern to retrieve information (other MBSE tools has similar capabilities). Context patterns is the description of a path of elements in a model through a list of tokens. Fig. 3-31 illustrates an example of a screenshot of a table showing the objectives and their indications with criteria descriptions. The table was retrieved by applying the following context pattern: *Package**, *{Objective}Objective, {Approach}has approach:,{Criteria Approach} allocate:*|*{Indication} Indication CB* | *{Indication} Indication cnst*| *{Indication} Indication action* | *{Indication} Indication Act, {Criteria} allocate:*.

In conclusion, this stage changes significantly compared to MOKA. It adopts from MOKA the MML product model. Nevertheless, a disadvantage found from MOKA's process model is that most of the knowledge had to be linked to the knowledge book. This step achieves a knowledge representation that favors on keeping the knowledge within the same modeling tool. The knowledge book is still consider important, especially for a quick knowledge capturing process. Nevertheless, todays technologies permit to move entirely from a document-based approach to a model-based approach understandable for any engineer. For the exceptions where the knowledge cannot be represented then a link to either the knowledge book or the knowledge source can still be maintained in the same modeling tool.



Name in Objective	Name in Approach	Description...	Name in Indication	Specification in Indication	Name in Criteria	Description in Criteria
CreateStarOfSlots	FollowStarOfSlotsSteps					
GetVoltageOnConstantTorqueCurve	FEMCoupledAlgorithm_VoltOnTorque		Algorithm10		Algorithm10D...	This algorithm finds the giv

```
1: Given Torque_ref, tolerance_torque>0, speed, Vref, angle_vector=[angle0,.., angleEnd] where angle0 < angleEnd, Initial Is0, k:=0, kmax,
step_number = n; motor model and tool parameters.
2: Initialize variables torque=[]; Is=[]; angle=[ ]; Fd=[]; Fq=[]; Vt=[]; Stop=False; i:=0
3: While not Stop
4:        run algorithm1 for anglei
5:        compute Vt with equation
6:        append values in torque, Is, angle, Fd, Fq, Vt
7:        if Vti<= Vref: Stop = True; Optional After Stop apply bisection method as needed between anglei and anglei-1 particularly when
angle_vector steps are large.
8:        if i==len(angle_vector)-1: break;
9:        i=i+1
10:Apply cubic spline interpolation to variables Is, angle, Fd, Fq, Vt
11:Get values at Vref
```

Name in Objective	Name in Approach	Description...	Name in Indication	Specification in Indication	Name in Criteria	Description in Criteria
⊟ ReduceCoggingTorque						
	Geometry		ApplySkewAngle	Apply skew angle to rotor or stator		
	Geometry		LstkVSDso	Increase Lstk rather than Dsi because Tcogging(Lstk) < Tcogging(Dsi) for all Lstk = Dsi in per unit	LstkVSDsoR...	Tcogging is proportional tc
	StatorAndRotorTopologyTC		QsAndPRelation	Qs=2p+-1; the more pole pair the lesser cogging torque amplitude	QsAndPRelat...	These machines are asym
	StatorAndRotorTopologyTC		mcm_QsNpN	The higher the LCM(Qs,2p) the lower the cogging torque		
⊟ ReduceStructuralVibrationsAndNoise						
	AerodynamicsAirstreamSirenEffectAndFans_Structural		AirDucts	Eliminate air ducts to avoid the siren effect. If not possible find proper place (siren effect).		
	AerodynamicsAirstreamSirenEffectAndFans_Structural		ChangesOfFlow	Eliminate abrupt changes or discontinouties in flow directions whenever possible (Airstream).		
	AerodynamicsAirstreamSirenEffectAndFans_Structural		CrossSection	Make large flow path cross sections if possible all over its length and/or make gradual changes in the cross sections (Airstream).		
	AerodynamicsAirstreamSirenEffectAndFans_Structural		DirectionalFan	Use directional fans (Fans).		
	AerodynamicsAirstreamSirenEffectAndFans_Structural		FanBlades	Choose the number of fan blades properly.		
	AerodynamicsAirstreamSirenEffectAndFans_Structural		FanClearence	Increase the cleareance between the stationary parts and the fan using non-symmetrical group of blades in relation to the stationary protrusions (Fans).		
	AerodynamicsAirstreamSirenEffectAndFans_Structural		FlowPath	Apply short streamlined flow path in order to have air motion in an orderly and predictable way (Airstream).		
	AerodynamicsAirstreamSirenEffectAndFans_Structural		SmoothSurface	Make smooth boundary surfaces (Airstream).		
	AerodynamicsAirstreamSirenEffectAndFans_Structural	It is usually	Obstructions	Avoid unneeded obstructions in the flow path (Airstream).		
	Bearings_Structural		BallBearingsFriction	Avoid friction in ball bearings by using thrust washers for axial preloading of bearings and/or select proper type of grease.		
	Bearings_Structural		Endbells	Avoid endbells natural frequencies match ball bearing characteristic frequency.		
	Bearings_Structural		FrameTolerances	Reduce shaft or frame tolerances.		
	Bearings_Structural		OilGrooves	Avoid instability by changing bearings grooves and/or change oil viscocity.		
	Bearings_Structural		SleeveBearingsFriction	Avoid friction by increasing the clearance, oil viscocity.		
	ShaftAndCoupling_Structural		ShaftMisalignment	Select preferably flexible couplings to avoid shaft misalignment with the driven shaft.		
	StatorAndRotor_Structural		ApplySkewAngle	Apply skew angle to rotor or stator		
	StatorAndRotor_Structural		Inertia	High inertia filters torque ripples, having less speed ripple therefore less vertical accelerations. The larger the diameter the higher the inertia.		
	StatorAndRotor_Structural		LooseLaminations	Assure good clamping of laminations.		
	StatorAndRotor_Structural		UnbalanceRotor	Assure good rotor balancing		
⊟ ReduceTorqueRipple						
	Electrical		AmpereTurns	Find optimal value of ampere-turns by deriving the studied harmonic amplitude of the torque with respect to ampere-turns.	TorqueRipple...	We can also have the cas
	Geometry		ApplySkewAngle	Apply skew angle to rotor or stator		
	Geometry		LstkVSDso	Increase Lstk rather than Dsi because Tcogging(Lstk) < Tcogging(Dsi) for all Lstk = Dsi in per unit	LstkVSDsoR...	Tcogging is proportional tc
	StatorAndRotorTopologyTC		QsAndPRelation	Qs=2p+-1; the more pole pair the lesser cogging torque amplitude	QsAndPRelat...	These machines are asym
	StatorAndRotorTopologyTC		mcm_QsNpN	The higher the LCM(Qs,2p) the lower the cogging torque		
	StatorAndRotorTopologyTr		Kw	Select a slot-pole combination satisfying a low winding factor (Kw) amplitude for the most conflicted 6k+-1 harmonic.		
	StatorAndRotorTopologyTr		SelectPolesAvoidingNaturalFrequencies	Select number of poles avoiding resonance i.e. taking into account any frequency of the torque ripple match a natural frequency of the application mechanical system.		

Fig. 3-31. Screenshot of knowledge retrieval example through relational tables.

3.4.5 Package

This phase emphasizes the knowledge implementation and integration during the software development. In this stage, the KBE applications development and integration with other entities take place. In other words, the software system is developed. Usually, the main goal to achieve in a KBE project are KBE applications for both the macro and micro level.

In order to develop a software system with these characteristics three options are available. Option number one is to start from scratch and develop the software entirely. Option number two is to engage with a software supplier company and make a collaboration agreement with them to develop the KBE system by customizing their applications and developing new features if required to achieve the desired system. For instance, ANSYS and Siemens are companies that offer a complete suite of software applications that can be integrated to achieve the wanted KBE system. Option number three is to develop the KBE applications and integrate them with commercial off the shelf software (COTS). The first mentioned option is a long-term development process approach and less likely to be considered, unless you are interesting to introduce a new software to the market. The second option creates complete dependency on an external company but could be an approach worth of consideration.

This thesis implements the third option. This type of implementation can be undertaken within the same company, reduces the development time and provides the option of not changing radically the process the eMachine experts carry out by reusing the same design tools they currently used. In this way, this work suggests a top-down decomposition of the software system and developing using a bottom-up approach for its construction. Therefore, an approach that fits to this type of system implementation is component-based software engineering (CBSE) [122]–[126]. CBSE principle relies on coupling independent components into systems.

Sommerville in [124] combines the definition of component from Heineman [126] and Szyperski [125] and defines it as: “A software unit whose functionality and dependencies are completely defined by a set of public interfaces. Components can be composed with other components without knowledge of their implementation and can be deployed as an executable unit”. Therefore, the main point is to integrate the selection of commercial software components with the KBE applications to accomplish the goal for the product development process. Moreover, the software engineer must develop the micro-level KBE applications as components for reuse. The COTS (CAD/CAE, requirement engineering, PDM) defined for the KBE system typically provide access to their application through ActiveX, APIs and/or plug-ins, which may require implemention for different programming languages. Then, the construction of middleware such as adapters is crucial to undertake the proposed approach.

Focusing on KBE applications, this work decomposes them in three parts, the knowledge wrappers, auxiliary modules, and the user interface. The knowledge wrappers are the software modules with no access to edition to the end-user that contains (wraps) the formalized knowledge from the previous step. After modeling the knowledge using the representation described in the previous section the knowledge engineer and software engineer must work together in order to convert that knowledge into code. Having the software modules that wraps the knowledge classified from the rest of modules facilitate the KBE application upgrade and maintenance. The auxiliary modules are software modules that makes the KBE application run properly but does not contain the formalized knowledge.

The user interface (UI) is the set of commands or menus on which the end-user will interact with the KBE application. Preferably, the UI should have the capability of visualization (diagrams, instructions, warnings, etc.) of the integrated knowledge to the end-user, when not possible at least handle the documentation containing the integrated knowledge. From the experience gained, the author suggests to develop micro-level KBE applications with no graphical user interface (GUI) whenever possible but provide a user interface through commands in a language the end-user may be familiar with. This increase the development process and allows the end-user experience with the KBE application faster. For the macro-level, GUIs may be more appropriate to be implemented.

In conclusion, the package stage is in general the software development process executed by the software engineer with the support of the knowledge engineer. The package name highlights the translation of the formalized knowledge into code (kept in modules) for the KBE application. The modules that contains the formalized knowledge are called in this thesis knowledge wrappers. However, in order to build a KBE system other software components may be involved, therefore, middleware has to be develop in order to execute the KBE system properly. A detailed description of the software development process is out of the scope of this thesis. This works centers more its attention on the knowledge integration and execution of KBES in the eMachine development process with emphasis in the design process. The software engineer should choose the methodology to follow and he may support the software development process with Application Lifecycle Management tools (ALM).

3.5 Discussion

MOKA methodology was adapted and extended for the electric machine design process. The purpose of this adaptation and extension of MOKA is to facilitate its implementation in the electric machine industry. The capture and formalize steps provides sufficient details for the Knowledge Base generation. The knowledge base is the foundation of a KBE system.

Moreover, the proposed adaptation and extension of MOKA for electric machines cope with the shortcomings mentioned in the introduction in the following manner. The first limitation says MOKA focused more on the knowledge engineer and not the end user. In this work, insights for the identify and justify phases centered in the electric machine developer were given to support the decision and analysis of the application of a KBE system as a solution to the enterprise needs. The second limitation highlights the lack of supporting tools and mechanisms of knowledge representation, for this reason, this chapter gave example of implementation options by using Microsoft Excel for the informal model and MBSE tools as a KB tool for the formal model. Regarding the third shortcoming of not considering its usage in the design process and knowledge reuse and maintenance a solution to this problem is addressed in the next chapter.

The fourth and final shortcoming is that MOKA is more product-oriented and not process-oriented. In this thesis it is deduced that the product model of MOKA is well structured and well defined that its adaptation to any application is simple. Therefore, examples of the product model applied to electric machines (eMachine product model) were illustrated to serve as reference of its usage to the reader.

Problems were encountered to represent knowledge using the process model specially by not having some knowledge forms available in the same KB tool. For instance, the need of linking an activity to an informal rule form instead of having the rule embedded in the same diagram or at least in the same software tool. In addition, MML only focus on the micro level view no knowledge representation for the macro level view is provided.

Hence, a new proposal of knowledge representation was developed for the eMachine process model. This was divided into micro-process and macro-process levels. For the macro-process level SysML was adopted as the language for this case. Example of the macro-process knowledge representation is the proposed requirement-engineering framework (presented in Chapter 2) addressing the process from the initial specifications provided by the integrator and deriving them until lower level requirements are obtained to begin the design process. For the micro-process level, a new metamodel with stereotypes extended from SysML classes was provided and described through several examples. This metamodel is considered a contribution of this thesis. Apart from

SysML a new profile called KBeMotorProfile was created and described with the stereotypes it contains. Examples of micro-process knowledge representation using the new profile were illustrated as reference to the user.

The package phase was introduced in this chapter providing high-level insights of the phase. The detailed description of software implementation is out of the scope of this work but it is an important part of the entire process.

In conclusion, substantiation of the Knowledge Base generation process upon the development of KBE systems for the electric machine industry was described. A new knowledge representation for the eMachine process model was derived providing examples of each kind of knowledge representation.

Chapter 4

KBE SYSTEM FRAMEWORK

This chapter introduce the KBE system framework for the development of electric machines. The framework integrates and adapts relevant technologies used in industry for the product development process. The proposed framework synthesizes the concepts presented in the previous chapters in order to create a successful KBE system. The framework is based in the so-called KBV2-model, which consist of the Knowledge Base generation process adapted from MOKA methodology and the eMachine standardized macro-level framework. In addition, the actors, tools, knowledge management, and KBES SW architecture is presented.

This chapter is divided in seven sections. Section one presents an introduction to the framework. Section two introduces PLM and PDM. Section three describes the KBV2-model with the generalization of actors, and tools. Section four presents the knowledge management features by describing the knowledge base structure and data management. Section five shows the proposed architecture for the KBE system SW. Section six presents an implementation example for the eMachine Product Model. Finally, section seven discusses the results.

4.1 Introducción

Contemporary technologies relevant in industry were reviewed and described in the previous chapters. The proposed solution is to use and integrate these technologies to achieve a complete KBE system for the development of electric machines. Fig. 4-1 summarizes the involved technologies and their integration is described in this chapter.

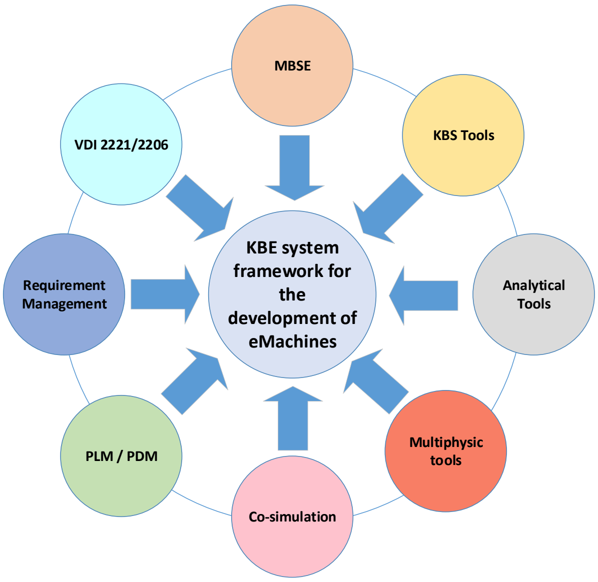


Fig. 4-1. Integrated technologies to develop a KBE system.

Chapter 2 presented and suggested a standardized macro-level framework concept to develop electric machines. Chapter 3 introduced the Knowledge Base generation process and its application for electric machines including knowledge representation. Merging the Knowledge Based Generation process and the eMachine standardized macro-level framework adapted from VDI-2206/2221 is the core concept of the proposed KBE system framework. This chapter synthesized the concepts so far explained giving additional description and generalization of aspects such as actors, tools, knowledge management and architecture.

Moreover, an implementation of the eMachine Product Model is presented at the end of the chapter. The use case example is metadata customization of PDM systems with the goal of further use in the implementation of the development of a KBE system presented in Chapter 6.

4.2 Product Lifecycle Management and Product Data Management

This section clarifies the concepts of Product Lifecycle Management (PLM) and Product Data Maganement (PDM). Product data is all the data in the process that is used to imagine, design, produce, use, support and dispose the product. According to [127] p. 134, product data management "is the activity of managing product data". PDM systems were introduced in the early 80s, due to its usage in today's industry it has become one of the main software in product development. A PDM system is a software application that manages product data and is part of a PLM application [127]. PDM system is a crucial part of a PLM application and its selection and implementation has been unclear in industry. People still struggle about the scope of a PDM system there is a misconception in the concepts of PLM and PDM, this will be clear in a concise way in this section from the information gathered from literature.

PLM comes to integrate the business activities by organizing and making information and processes available throughout the lifecycle of the product in a collaborative platform. Stark [127] p.1 defines PLM as "the business activity of managing in the most effective way, a company's products all the way across their lifecycles; from the very first idea of the product all the way through until it is retired and disposed of". Schuh et al. [128] p. 210 define PLM as "a systematic concept for the integrated management of all product related information and processes through the entire lifecycle, from the initial idea to end-of-life".

On the other hand, PDM centers on managing design data of the product development processes. PLM focus on managing the whole process related to product lifecycle. Fig. 4-2 shows the lifecycle of a product and the complexity of the whole process, the vertical axis shows the components that need to be addressed to manage the whole process and the horizontal axis the lifecycle phases.

Solidworks [129] p. 2 states "PDM is a design-focus technology that increases efficiencies within existing product development processes by improving the management of product design data. PLM on the other hand is a strategic, process-centered approach that leverages PDM and other technologies, along with consulting services-to manage product lifecycles, remake processes and increase output. As a result PLM improves productivity across the connected enterprise rather than in a single department or specific process".

PDM is useful to transform raw data (drawings, datasheets, reports, etc.) into metadata, which provides more information about the data. It also helps query data easily retrieving past files that can be helpful for a new problem. PDM links all data to the belonging product in a collaborative environment allowing availability of information worldwide 24/7. PDM guarantees an improvement in reuse of design information, in use of resources, access to files, control engineering changes, support of customers and security of information with the profit of reducing lead time and costs [130], [131].

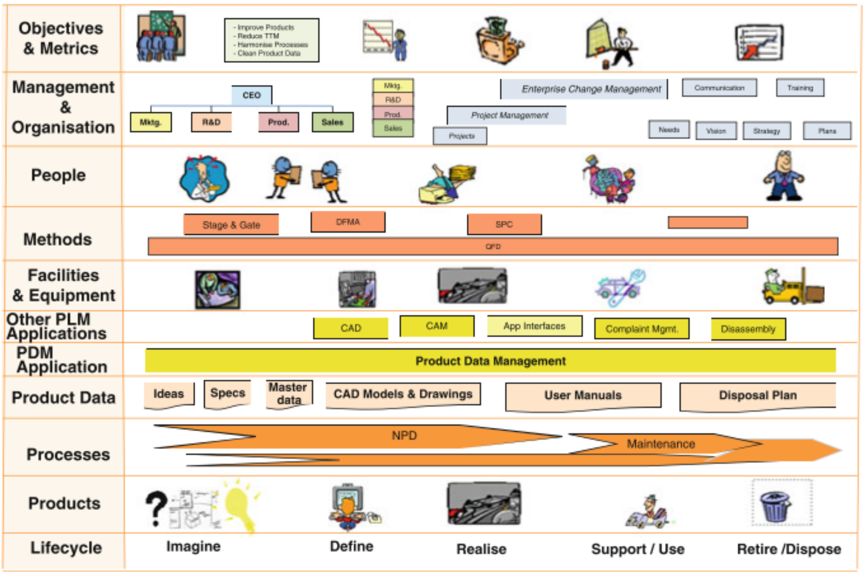


Fig. 4-2. Product lifecycle management components and phases [131] p.5.

Selecting the ideal system requires a complete evaluation of the needs, goals, and structure of the product development and manufacturing organizations, a guideline of PDM and PLM selection and implementation can be read at [127], [128]. Depending the size of your company, you may need a PLM or PDM solution. Fig. 4-3 presents what PDM and PLM covers as well as the lifecycle stages of a product, it can be seen that PDM may have part or all of application integration, visualization, collaboration and reporting and analytics.

Finally, some examples of commercial PLM tools are:

1. Siemens Teamcenter
2. PTC Windchill
3. Autodesk Fusion Lifecycle
4. Oracle Agile
5. Arena PLM
6. Dassault ENOVIA
7. SAP PLM
8. Omnify Empower PLM
9. Aras Innovator PLM

Other tools only offer data management such as SYNECT from dSPACE.

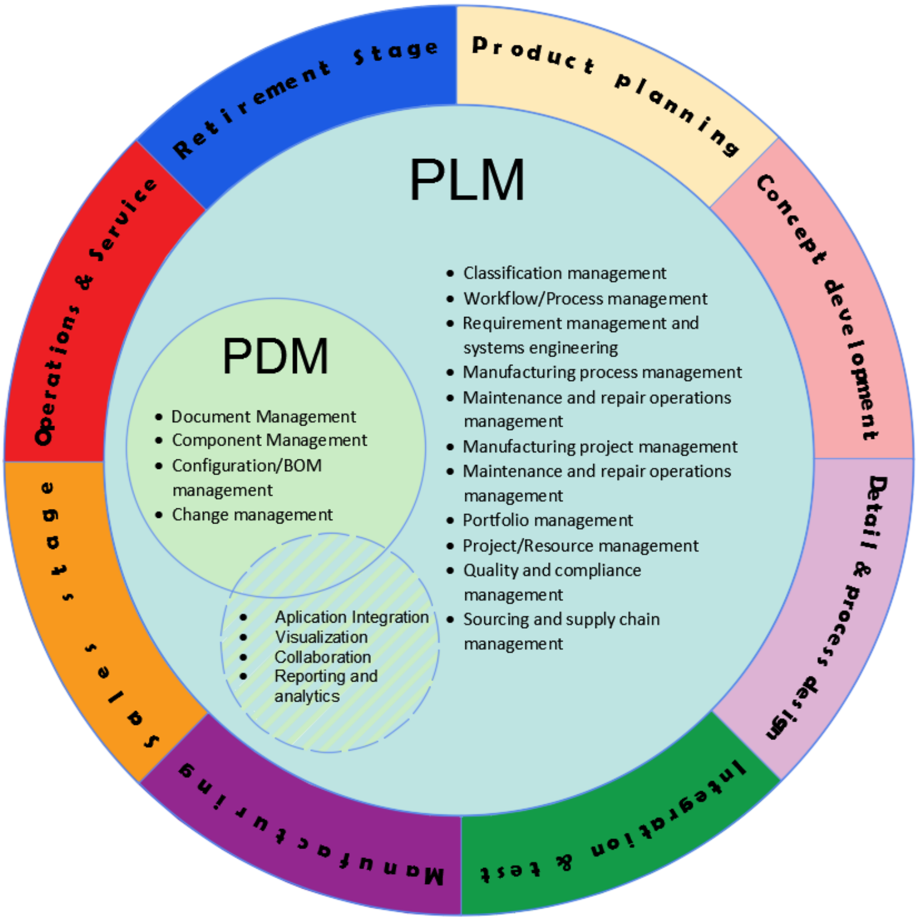


Fig. 4-3. PLM and PDM relationship and lifecycle stages adapted from [4], [130].

From literature, it can be concluded that less time should be spent on deciding which PDM or PLM system to purchase instead the software should be adapted to the company needs and by working this way its implementation would be fast enough to start getting profit out of it. This thesis follows this philosophy by selecting and open source PLM and customize the metadata to fulfill the objective of this work. Therefore, the selected PLM software is Aras Innvator. In addition, based on its availability the data management tool SYNECT is also used. However, during the implementation of the KBE system framework only PDM features of Aras Innovator will be used since the implementation scope of this thesis is limited to an electric machine department. More details about both tools will be given later in the respective implementation section.

4.3 KBV2-model

The proposed KBE system framework is encapsulated in the KBV2-model. Fig. 4-4 shows the KBV2-model that stands for Knowledge Base generation for the V2-model. This model synthesized the processes described in Chapter 3 and Chapter 4. This model is composed of two main parts, the Knowledge Base generation process (left section) and the standardized macro-level framework to develop electric machines (right section). The model is devised to provide the involved actors a holistic view of the process flow for the development of KBE applications regarding the electric machine context. At the same time, their responsibility on the process. In other words, when implementing the model for a specific KBE project (for either the macro or micro scale), the model should reflect the KBE application purpose for the V2-model. Schemes can be added to clarify the KBE application purpose into the standardized macro-level framework. Working in this form introduces the systems engineering ingredient for the development of a KBE system showing KBE applications integration with other entities of the eMachine development system. Chapter 3 described the standardized framework, which is created following the VDI guidelines to facilitate its adaptation to any current methodology an enterprise uses.

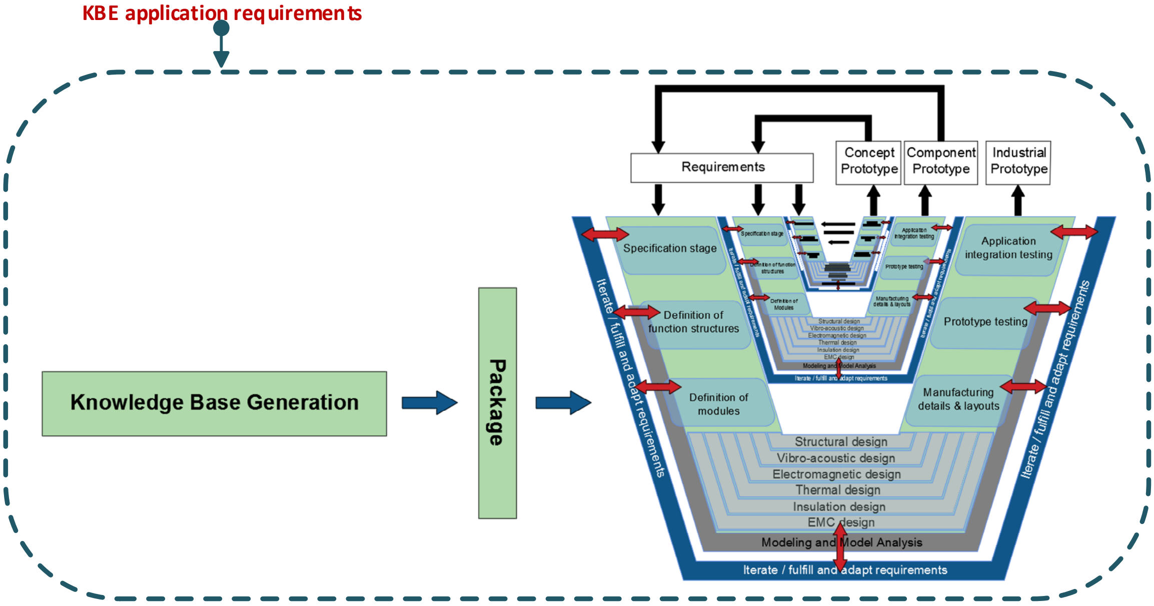


Fig. 4-4. The KBV2-model.

The previous chapter mentioned one of the shortcomings MOKA has is the no attention of the KBE application usage in the design process. Then, including the eMachine standardized design processes overcomes this issue by providing the actors a general view to specify the purpose of the KBE application into the system of the company.

The model intends to serve for both the macro and micro levels. However, the scope of what is micro and macro should be defined. The Knowledge Base generation process maintains its structure for the macro and micro levels. The only difference lies on the knowledge representation. What defines the macro level perspective is the standardized macro-level framework, the actors must agree on the requirements of the KBE application regarding the entire system or only part of it, for instance only one V-cycle or particular high-level steps of the V-cycle. Taking into account the tools, stakeholders and interchange of information among these. For KBE applications in the micro scale level the particular process where it wants to be deployed ought to be specified. The aim of the micro level view is to create KBE applications for particular processes within the seven steps of a V-cycle. It is important to remark that the micro level view is the most adopted approach for developing KBE applications [13], [29], [121]. Another way to view the difference between macro and micro could be in terms of the system outputs, for the macro-level the suite of KBE applications are part of a system to support the product development process ending up with a product concept, component or industrial product for mass-customization. The micro-level case supports the fast execution of a process for a particular task. The actors must agree and define the boundary between the macro and micro level for the KBE project.

Introducing the MVP principle (Appendix A) allows the end-user integration to the KBE application development process. The main idea is to develop KBE applications following this principle. In particular, the type of MVP that must be pursue is the single feature MVP [132]. To make this possible the macro level KBE application, which would be the ultimate goal, should be broken down into micro level KBE applications (components). During the implementation of this framework, this proved to be helpful for not only fast developing and integration but also to gain feedback from the end-users allowing the improvement of features for every KBE application. Fig. 4-5 illustrate the MVP principle to create a macro-level KBE application. The figure tries to show how KBE applications with less features can be extended by developing in multiple facets until the desired KBE system is achieved.

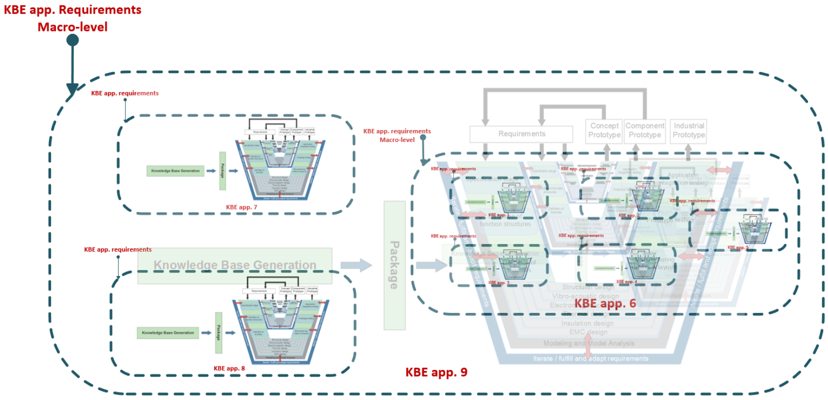


Fig. 4-5. MVP principle for multiple facets development of a Macro KBE application.

This principle could influence general managers not assimilating KBE as a solution to a problem, which could be an obstacle to submerge into a KBE project. A feasible alternative in case of economic uncertainty is to develop a MVP to see if it is a real solution to the enterprise needs. One aspect to highlight is KBE relies on well-known knowledge since its goal is to automate routine tasks. Therefore, the technical uncertainty to achieve a goal is low. Another advantage of developing in multiple facets is that the end-user can leverage the benefits from single features applications faster rather than waiting for a finished viable application.

Moreover, the process to develop an electric machine in a high-level form (as shown in the eMachine macro-processes) is similar for any application domain. Application domain in this context means the particular field where the machine accomplish its final purpose i.e. integration (automotive, elevator systems, aerospace). The difference among the application domains stands on the requirements the machine satisfies deriving in different design features in structure and performance. Thus, going deeper into details, the knowledge differs from application domain to application domain although the design domains maintains (EMC, insulation, electromagnetic, thermal, vibro-acoustics and structural). For this reason, the proposed KB structure in this framework takes into account this issue by creating different units in the KB considering different electric machine applications, this is explain in more detail in Section 4.4.

The sunburst diagram in Fig. 4-6 summarizes the proposed KBES framework. The planning and break down of modules should be carried out from the innermost to the outermost (same as a top-down approach). Each V-cycle step can be decomposed into more sections. The developing process following the KBV2-model should be executed from the outermost to the inner most (same as bottom-up approach).

In summary, the main purpose of the KBV2-model is to support the actors by providing a holistic view of the process to facilitate their decisions on the aspects they muset take into account in order to develop the KBE system. The economic feasibility of a KBE project is not consider in this model. It is only focused on the technical aspects of KBE. The eMachine standardized macro-process and the Knowledge Base generation process. A synthesis of the involved actors in the KBV2-model and tools generalization are described in the following subsections.

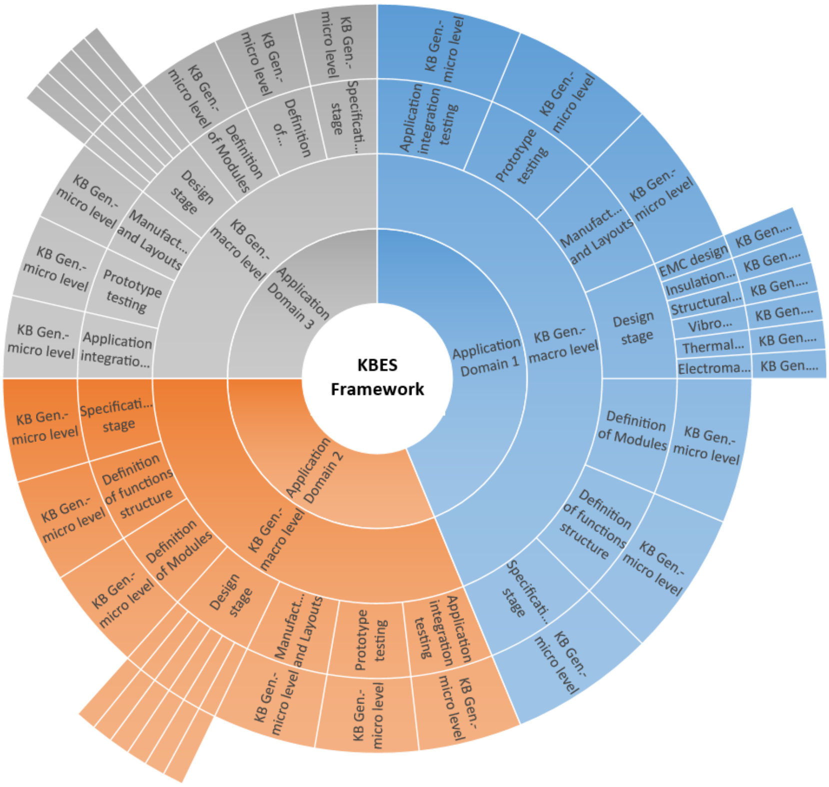


Fig. 4-6. KBE System framework for the development of electric motors.

4.3.1 Actors

The proposed KBES framework identifies five actors regarding the technical aspects of a KBE system. These actors are the knowledge engineer, software engineer, systems engineer, eMachine experts, and end-users. One person may play different roles in the KBE system framework. The use case diagram in Fig. 4-7 presents the main roles of each actor in the KBES framework. One of the limitations of this framework is the not inclusion of managerial processes guidelines during the KBES development, for instance, resources management, communications guidelines among actors, cost analysis and risk analysis for KBE projects. Then, let us focus on the technical aspects of each actor in the KBES framework. The actors are referred in singular form, nevertheless, the diagram shows that for each actor one or more people may be involve.

4.3.1.1 Systems engineer

This framework introduces the systems engineer. The system engineer holds the holistic view of the eMachine development process, thus, he has the higher hierarchical level among the rest of actors. Due to its profession nature, this actor is responsible of making or analyzing (in case he did not give the proposal) the formal proposal request for a KBE solution in order to improve the efficiency of the current system. Definitely, this cannot be done without analyzing the needs and taking into account the opinion of the eMachine experts (the actor providing the knowledge for the KBE solution). Typically, the idea of KBE arises from the need of actors with lower hierarchy, for example, the eMachine designers.

Once the project starts. The system engineer envisage and designs the KBE system. The KBE system, in summary, is the system describing the integration of the KBE applications and other software components into the product development process. Therefore, he is in charge of showing how the suite of applications will interact with other entities of the current system. This includes the definition of requirements for the KBE application with the support of the eMachine developer and end-users. Another task to carry out is the definition of a standard information exchange among tools and actors. For instance XML among software, xlsx to present results to the end-user, SysML among actors. He also should break down the KBE project following the MVP concept as shown in the sunburst diagram in Fig. 4-6. The systems engineer assesses the entire project by monitoring the KBE system development. Finally, he must evaluate the performance of the finished KBE system in the product development process.

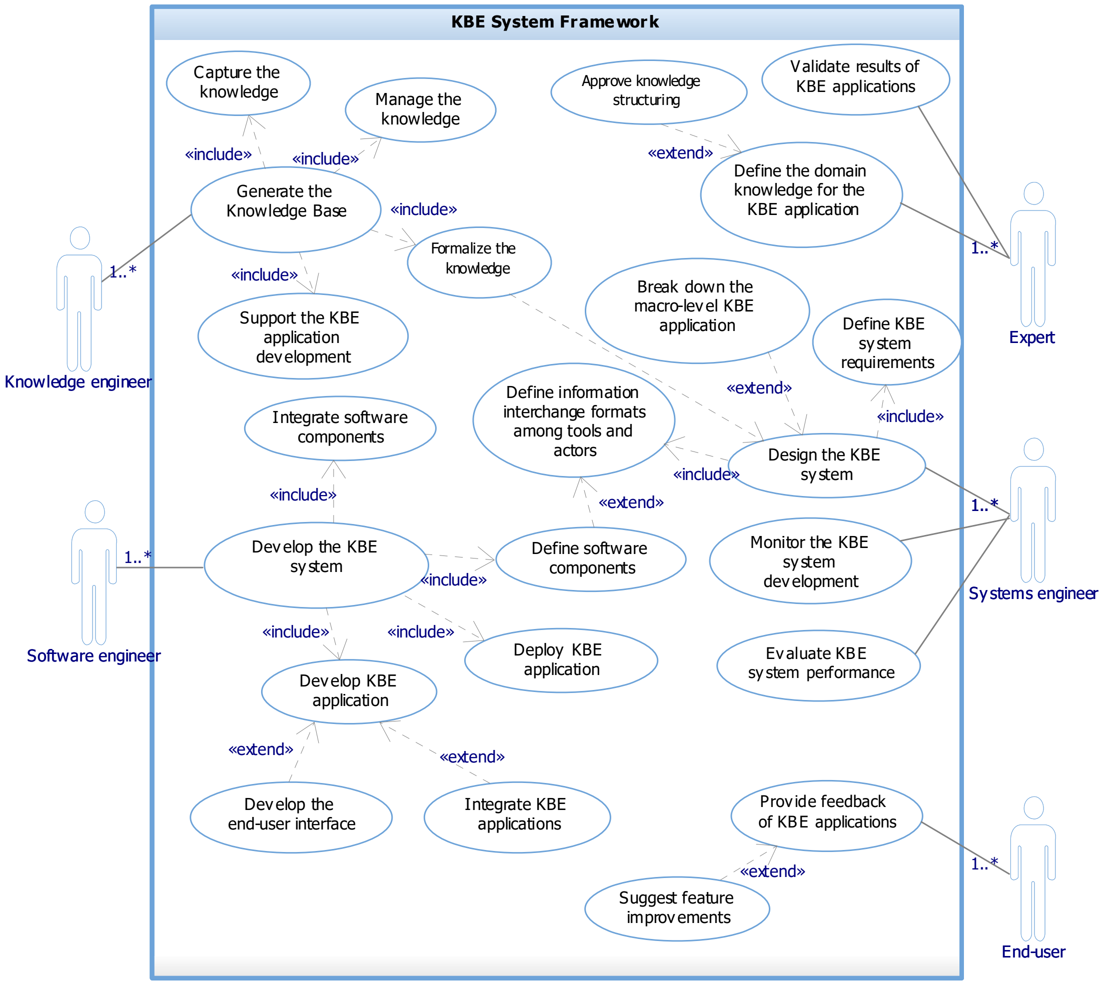


Fig. 4-7. Actors and their roles in the KBE system framework.

4.3.1.2 Expert

The expert can be any of the eMachine developers. The generalization of the eMachine developer was presented in Fig. 2-13. The electric machine developer is the actor responsible for the design/manufacturing of the machine satisfying the requirements of the customer. In the KBES context, he must define the knowledge for the KBE application, approve the knowledge formalization, and validate the results of the KBE application.

4.3.1.3 Knowledge engineer

This actor is in charge of the knowledge base generation, which includes the steps of capture and formalize. Therefore also responsible to manage the knowledge during the entire KBES development. He is responsible of the Knowledge Base generation process described in Chapter 3. Furthermore, he supports the knowledge implementation process that leads to the final KBE application.

4.3.1.4 Software engineer

The software engineer is the one who transforms the knowledge into a real functional software tool. He decides the development platform and the approach to achieve the system. He develops the KBE application, if several KBE applications as well as other software components are involved is capable of their integration as one software system. He also develops the user interface and is in charge of the KBE application deployment or KBE system deployment.

4.3.1.5 End-user

The end-user, typically more than one, is the operator of the KBE system. He is the specific eMachine developer using the KBE application. He can provide feedback including feature improvements for the KBE application.

4.3.2 Tools

The proposed KBES framework intends to integrate the technologies illustrated in Fig. 4-1 to enhance the product development process (PLM/PDM, KBS, MBSE, multiphysics, analytical, and co-simulation tools). Fig. 4-8 remarks where the technologies could be implemented on the KBV2-model. Nevertheless, their implementation is flexible and must adjust to the needs of the enterprise or design departments.

In order to generalize the KBES framework to a certain extent, this section provides a list of tools options that can be selected for the implementation (package) phase for a KBE project. A summary of state of the art commercial and open source tools are listed in Fig. 4-9. The purpose of providing this list is to expand the possibilities of implementation of the proposed KBES framework to the readers.

Even though in this thesis the implementation goal is accomplished by developing a KBE system regarding coding of KBE applications through programming languages. There are different ways of knowledge implementation. For instance, you could define an application only to inferred knowledge from your KB through inference engine tools (e.g. Protégé, PyKE, Prolog, and Lisp). Another way of knowledge implementation could be the creation of a product model for PDM metadata customization. Thus, the KBS tools herein provided can be used for several ways of implementation. However, it is remarked that the Knowledge Base generation process must stick to the steps and knowledge representation provided in this work where a MBSE tool is used as a KBS tool i.e. still using the knowledge representation shown in Section 3.4.4 to formalize the knowledge. The resulting KB could later be implemented as the actors wish e.g. in one of the listed KBS tools.

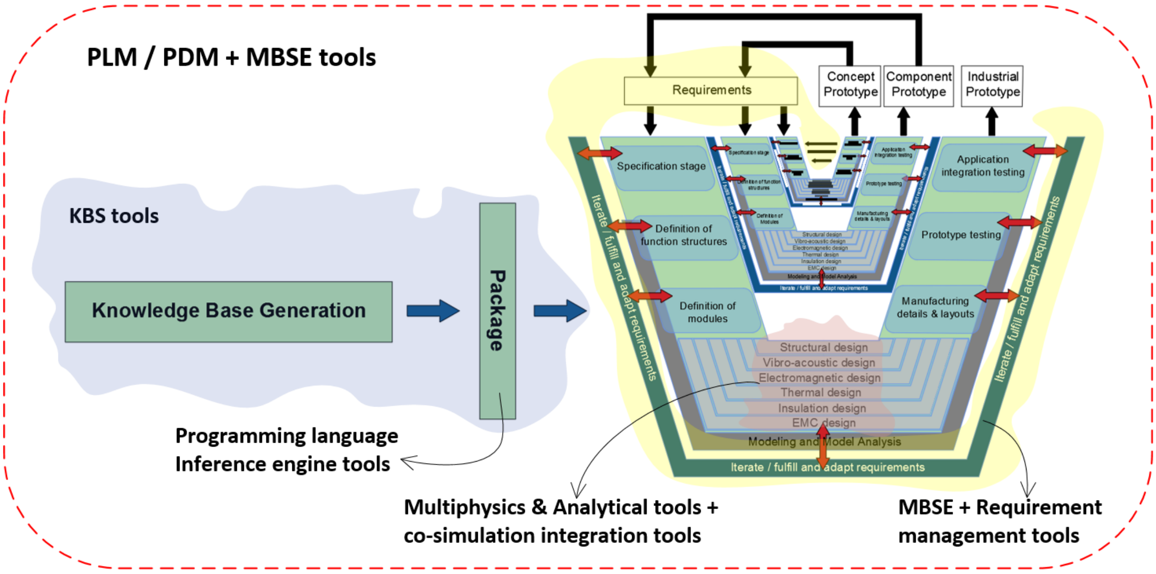


Fig. 4-8. Implementation of tools on the KBV2-model.

Moreover, MBSE tools should be used to support the design of the KBE system (i.e. applying systems engineering). MBSE is suggested to model the whole system in order to have an integrated way to trace and manage requirements in a cross-discipline environment as well as to have the entire process modelled to support the product development and risks could be detected from early phases. Requirement management tools with MBSE support the process of requirements traceability, satisfaction and verification for both the KBE system and the eMachine development process.

Multiphysics and analytical tools are normally used independently but lately the trend in industry is to work in an integrated way with a co-simulation environment (either commercial or custom software) whenever exist the need to achieve multi-disciplinary optimization (MDO). Finally, to manage the whole product lifecycle PLM/PDM is the top of the line tool to do so. In addition, it can be used as the source of truth in the KBE system.

MBSE tools + Requirement Managment tools
- No magic - Cameo System Modeler + Cameo requirements +
- IBM - Rational Raphsody + Rational DOORS
- Ansys - SCADE
- Vitech - Genesys
- Sparx Systems - Enterprise Architect
- Artego - Artisan Studio

PLM / PDM tools
- Siemens – Teamcenter
- PTC – Windchill
- Autodesk – Fusion Lifecycle
- Oracle Agile
- Arena PLM
- Dassault ENOVIA
- SAP PLM
- Omnify Empower PLM
- Aras Innovator PLM
- Solidworks PDM

KBS tools
- Prótégé
- MOKA ICARE + MML
- SysML / UML
- Technosoft - AML
- Genworks – GDL
- PROLOG
- LISP
- Jcolibri
- PyKE
- Jess for Java
- CATIA-Knowledgeware

Multiphysics and Analytic tools
- CAD/CAE tools
- Matlab / Simulink
- Maple
- Phoenix Integration - ModelCenter
- Altair-Hyperworks
- Ansys-Simplorer

Fig. 4-9. List of software tools available in market.

4.4 Knowledge management

To get the most out of the knowledge resources mentioned throughout this work. This section presents the knowledge-base (KB) structure and organization the author believes is the most appropriate for the successful management of knowledge in the KBE system. Furthermore, it presents the data and information organization among the different integrated technologies to the KBE applications.

4.4.1 Knowledge base structure and organization

The interest of the KBES framework is to develop KBE applications in multiple facets. Thus, the eMachine KB increases while the number of KBE projects increases. Fig. 4-10 shows the structure of the KB. The figure shows the eMachine KB consists of the generic KB and the application domain KB. Different applications domains (electric vehicle, elevators system, etc.) can be part of the KB. The KB is composed of units, every unit is part of the KB and a unit can be used by any specific project. The organization of the KB in units is derived from the implemented tool in this work, IBM rational Raphsody. A unit in Rhapsody is a file in which the modeled information is stored. Different types of units can be created in Rhapsody, however, only package units is of the interest to the concept of this thesis.

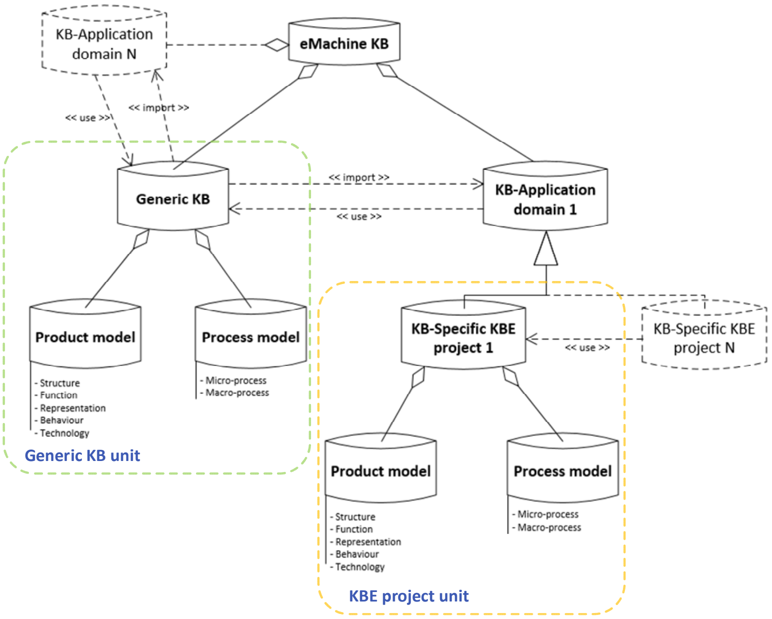


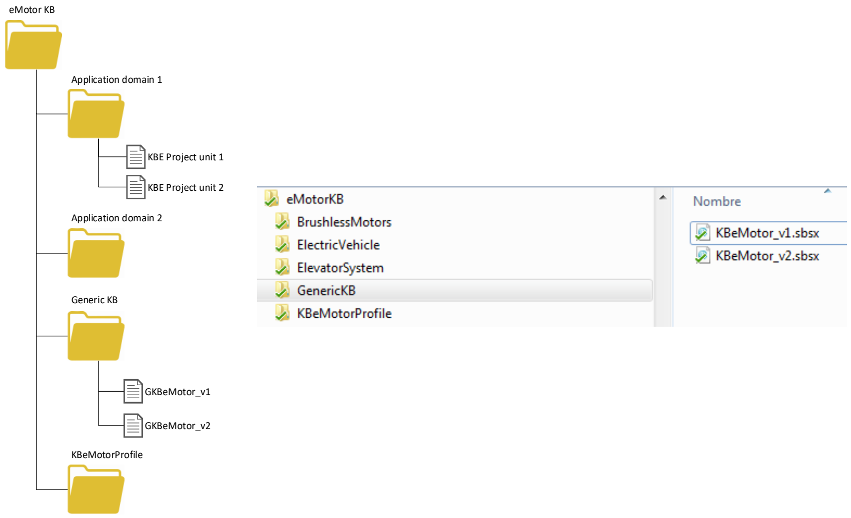
Fig. 4-10. Knowledge base structure.

Regardless of the selected tool used, for the interest of this work the unit is a file containing part of the formalized knowledge of the eMachine KB. It can be seen enclosed in the figure the two types of units, the generic KB unit as well as the KB unit for a specific KBE project. Every specific KBE project is a type of KB unit for a particular application domain.

This thesis recommends when beginning from scratch (no KB exists) that the starting point should be the creation of a generic KB preferably from the eMachine product portfolio of the company. This makes the knowledge engineer familiar with the knowledge domain and with this KBES framework. The generic KB should contained general knowledge about electric machines of the enterprise i.e. knowledge that can be used in a KBE project irrespective of the application domain. Therefore, the knowledge engineer with the support of the eMachine developer (experts) are accountable for the decision of what knowledge will be part of this KB unit. The generic KB is a finding that arose with the project of metadata customization of a PDM system, this implementation is presented in Section 4.6. It was found that this structure facilitates the reuse of the knowledge gathered and modeled in new KBE projects. Hence, the separation of KB units into specific KBE projects and the generic KB.

Every KB unit contains the product model and process model. The generic KB imports knowledge from specific KBE projects. A KBE project may use entirely or partly the knowledge withheld in the generic KB as well as the knowledge withheld in another KB specific KBE project from the same application domain type. Finally, it is important to clarify the generic KB is not a KBE project although it can contain complete KBE projects. Since these units derive from packages. Then, every unit stored in the repository can be imported as a package within the KB tool.

The organization of the units in the repository may follow a similar structure to the KB as illustrated in Fig. 4-10. To provide an example Fig. 4-11a shows the folder organization containing the KB units. Fig. 4-11b shows a screenshot of the generic KB folder containing two versions of units for the generic KB named KBeMotor_v1 and KBeMotor_v2. The file extension for the units in IBM Rhapsody is –sbsx–.

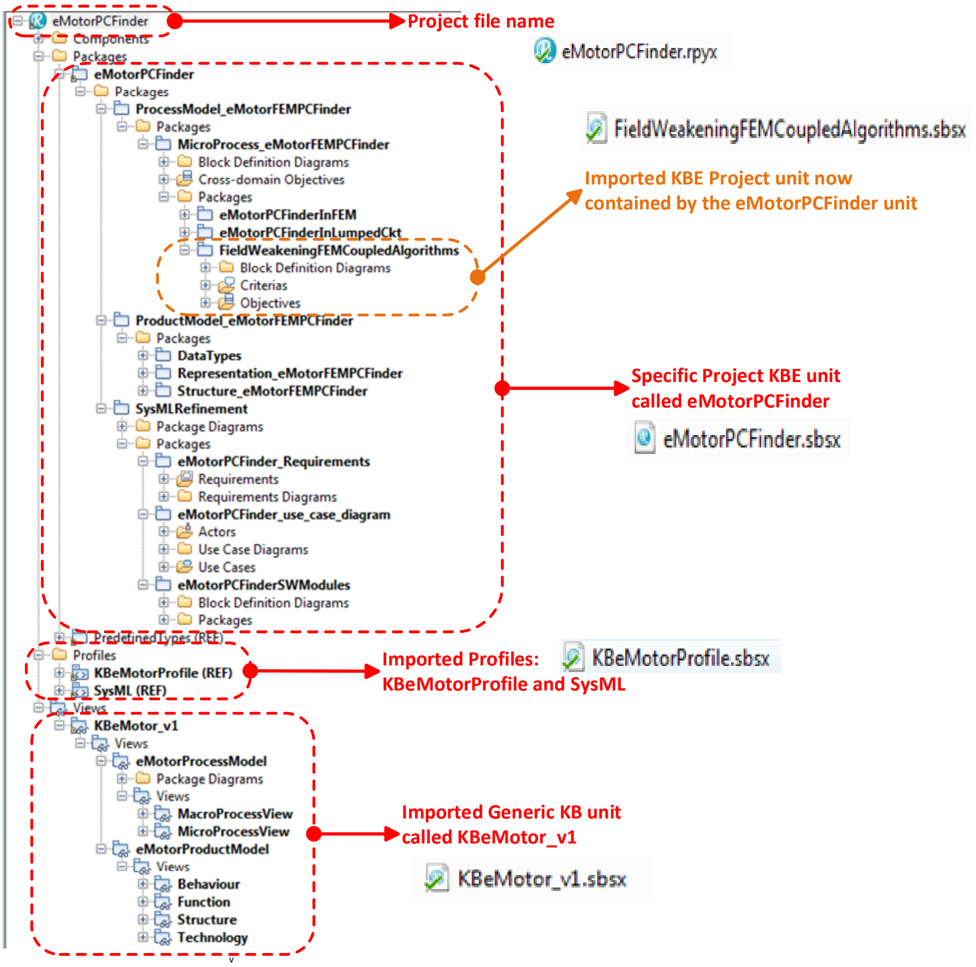


a) b)

Fig. 4-11. a) Folder organization of KB units and profiles. b) KB units organization example in a repository.

The new developed profile KBeMotorProfile for knowledge representation has the same file extension format (package unit in Rhapsody). However, in this work the KBeMotorProfile and SysML will be referred as profiles and not units.

The structure of a project in the KB tool is demonstrated in Fig. 4-12 (the structure can be applied in all MBSE tools). Within the project file, the units should be arranged as shown in the figure. On top, we have the specific KBE project unit (called eMotorPCFinder in the figure), followed by the profiles (SysML and KBeMotorProfile) and finally the generic KB unit (called KBeMotor_v1 in the figure). The same figure also gives an example of a KBE project using the modeled knowledge from a previous KBE project. This demonstrates the advantage of knowledge reuse by structuring the knowledge by units. That is, the units can be imported as needed for specific KBE projects.



Project file name

Imported KBE Project unit now contained by the eMotorPCFinder unit

Specific Project KBE unit called eMotorPCFinder

Imported Profiles: KBeMotorProfile and SysML

Imported Generic KB unit called KBeMotor_v1

Fig. 4-12. KBE project structure in Rational Rhapsody.

In conclusion, the KB structure is decomposed in generic KB unit and specific KBE projects units. The generic KB unit holds general knowledge from the product model and process model that can be reused by any application domain. All units may contain the product and process models, the organization of the units as files should follow a similar structure that allows the knowledge engineer to import any desired unit into a new project. The project organization within the KB tool was illustrated in Fig. 4-12. The order that should be followed from top to bottom is KBE project, profiles, and generic KB.

4.4.2 Data management

Data management refers to the administration of data in the product lifecycle. This work centers its effort in the design process. Thus, the integration of data is adapted to suit designers in the eMachine development process. Micro-level KBE applications are usually integrated in the same design tool (CAD, FEM, CFD), so the outputs from these are data files that can be stored in a shared or local repository. However, when dealing with macro-level KBE applications that are part of a KBE system, data management becomes a crucial aspect of the development process. From the perspective of a KBE system, this section provides an alternative for the organization of data in the eMachine development process. The reader must recall that the interest of the KBE system is to integrate a suite of KBE applications into the eMachine development process for a particular goal. Then, it is not the intention of the author to illustrate how to manage your data in a company but a proposal on how to manage the data regarding the creation of a KBE system to enhance the eMachine development process.

Lately, product development is executed in a collaborative environment. This approach allows knowledge share and data share among the stakeholders, in addition to the capability of design tools integration. PLM/PDM technologies trend is to integrate more tools, in many cases they still cannot fully support all tools capabilities integration, for example, MBSE and PLM integration is still an ongoing matter [4], [8]. For a designer, during the specific domain design process (electromagnetic, thermal, etc.) a more pragmatic approach is to work in a local environment. It is more common to find in literature KBE applications in a local environment. Nevertheless, due to the dispersion of design teams and manufacturers, either interdepartmental or around the world, keeping common data updated is crucial for a successful development of the product. An additional issue when working locally is that designers tend to manage files independently and when retrieval of data is required confusion may arise. Retrieval of important information may vary within a range from 30% up to 70% of the designers overall effort [133]. Therefore, an engineering data management system is relevant. Fig. 4-13 shows the architecture of data management from the KBE system perspective. The architecture shows a common data repository linked to the collaborative environment and to the local environment. The figure also shows the functionalities, it can be seen that it is in the local environment where the creation of design models and other files take place as well as modifications. The collaborative environment is managed through a PDM system. Section 4.2 described the advantages a PDM system can provide to the product development process. Therefore, from the KBE system perspective the end-user can have the capabilities of viewing models and files as well as to make changes whenever required through the integration of design tools e.g. CAD, CAE to the PDM system. Changes in this context means modifications to metadata information but depending the capabilities of the implemented PDM system changes may involve modifications to models in the same PDM system. During this research, it was found that even though PLM/PDM systems allows the integration of design tools many still lack of a built-in plug-in for many design software. Most of the design tools selected for the implementation of this work require a custom plug-in in order to make modifications to models within the selected PDM system. This is technologically possible but needs many hours of work. It is up to the actors to make the decision if modifications to models within the PDM system is worth of application. That is why the only functionalities suggested and highlighted for the end-user is to view and make changes to metadata information. Engineers' preference is to modify models or other files in a local environment for later saving in the repository and updating (filling or changing) the metadata information.

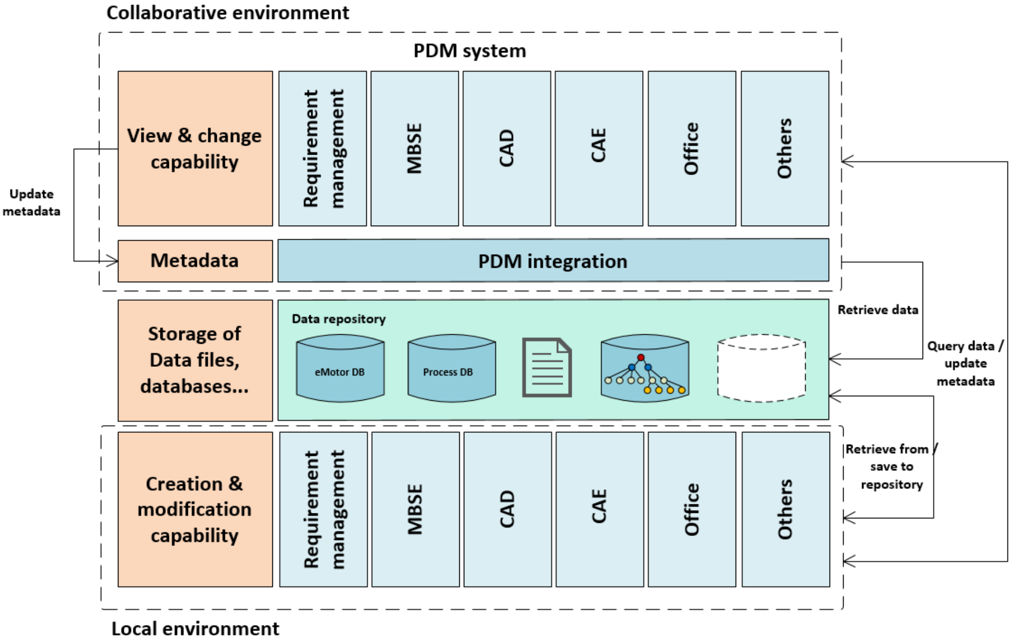


Fig. 4-13. Data management architecture.

Summarizing the functionalities regarding the KBE system, creation and modifications of models are managed locally; viewing of models and files as well as changes to the metadata information are managed in a collaborative environment. Both make use of a common data repository. The PDM system retrieves data from the repository, the eMachine developers can retrieve data from the repository but if these data is linked to the PDM system the metadata must be updated. The eMachine developer can save data to the repository and query data from the PDM system.

For the eMachine developers involved in the KBE system, it is advantageous to add to the KBE application the capability of accessing these data files directly from the repository (in a controllable way) and update automatically the changes in the PDM system for a collaborative work with other actors. This will reduce the workload of the designers and enhance the creativity work.

4.5 KBE system architecture

This section provides the architecture of the KBE system with emphasis on the KBE application. It was mentioned in section 3.4.5 that the KBE application is composed of the knowledge wrapper that consist of the formalized knowledge translated into code, the auxiliary modules that make possible the correct performance of the KBE application and the user interface. In order to integrate the KBE application to other software components middleware is mandatory. Middleware is the software that makes possible the interconnection among different software components enabling the applications to interface among them. Fig. 4-14 shows the architecture proposal for a KBE system.

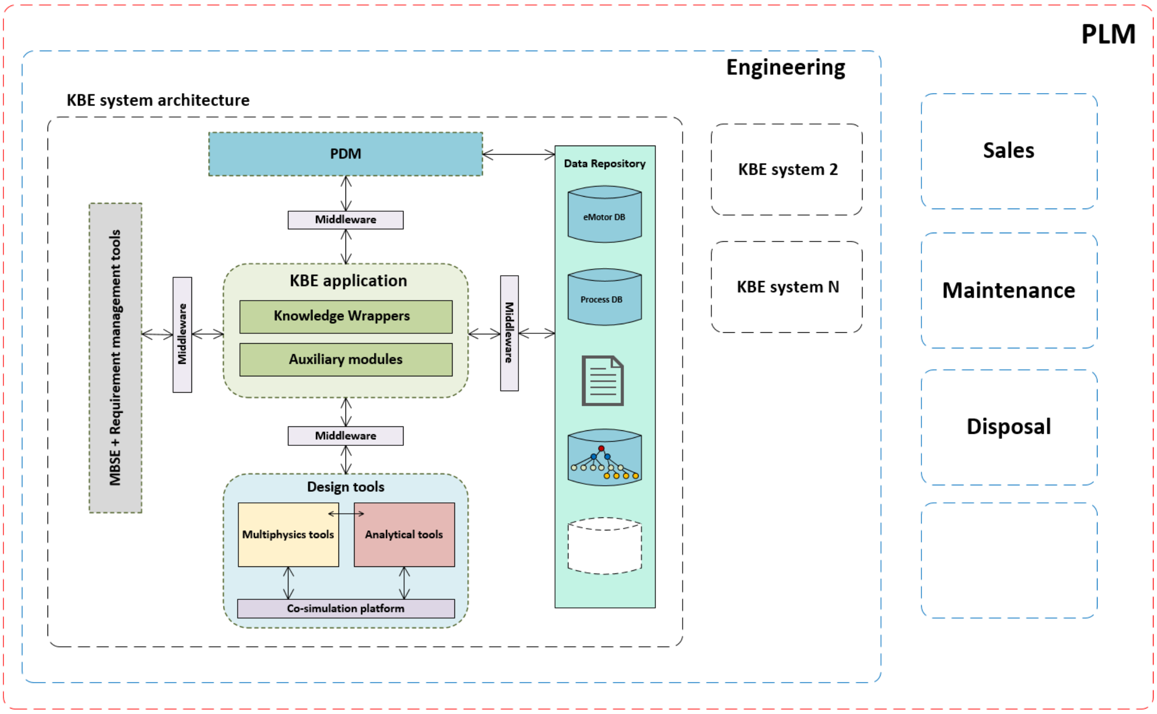


Fig. 4-14. SW architecture for KBE systems.

The KBE system consists of the integration of the KBE applications with the rest of technologies in the product development process. The architecture shows the KBE application communicates with the other technologies through middleware (see section 3.4.5). To cover the whole product development process the applications selected are PDM, MBSE plus requirement management, data repository and design tools. Nonetheless, the product lifecycle involves more processes not considered in this work. As a matter of concept, the architecture shows that a KBE system of the engineering department can interact with other departments by a PLM platform integration if required. The proposed KBE system architecture focus on the design process. The architecture is devised for the macro-level perspective. For the micro-level cases the developed micro KBE application may or may not require middleware for integration with software components, typically, their execution is embedded in the target software (CAD/CAE) or few links can be needed to other software components. The difference between macro and micro KBE applications can be seen in terms of their system outputs, for the macro-level typically the suite of KBE applications supports the product development process ending up with a product concept, component or industrial product for mass-customization, for the micro-level case it supports the fast execution of a process for a particular task.

Depending on the goals set by the KBE system, the eMachine development process can have one or several KBE systems. A common data repository is suggested for the KBE system. The architecture shows the design tools can be linked through a co-simulation platform but some design tools have built-in integration among third-party software that can leverage fast implementation. The KBE system architecture integrates the technologies for a successful product development process. However, the final structure depends on the devised system approved by the actors, the architecture is flexible enough to adapt to the needs in the product development process.

4.6 KBES framework use case example: PDM metadata customization

In this example, using the proposed Knowledge Base generation process an initial product model description is derived for a generic electric machine. The goal is to illustrate the eMachine product-model implementation by customizing a PDM metadata that will further be utilized for the KBE system presented in Chapter 6.

IBM Rational Rhapsody 8.3.1 is the software tool used for the creation of the KB. This section centers on describing the implementation of the product model i.e. the Package stage of the extended MOKA methodology for the development of electric machines.

4.6.1 Problem context

This work does not count with a given eMachine product structure metadata, therefore, one is created by capturing general knowledge from general machine structures and properties of each part. In this case, the developed generic KB intends to serve as a starting point to develop a product structure on a PDM software. However, if available, the enterprise can start the KB from their own product portfolio. The aim is to provide the reader with tools and steps that allow the process to be replicated. The generic eMachine KB holds general static knowledge. The purpose of this generic KB is to be the foundation for knowledge reuse for specific KBES projects regarding the design of electric machines. It is important to remark that the generic eMachine KB is not a KBE application but only knowledge that serves as a set of knowledge-tools basis to create any KBES application.

The implementation of the formalized eMachine product model described and shown in Section 3.4.4.1 is now carried out. The implementation does not undertake the development of a KBE application (coding). The knowledge implementation is applied into two commercial PDM software, Aras innovator and dSPACE Synect. In Aras all CAD related data is to be managed, thus, this application withholds the product structure metadata. On the other hand, Synect is more related to model-based design rather than CADs. So the metadata of Synect is customized to structure the data of electric machines multiphysics and analytical models.

4.6.2 PDM metadata customization

Next, the implementation is undertaken.

4.6.2.1 eMachine product metadata customization

Fig. 4-15 illustrates the knowledge transition from the KB to implementation. Starting from the KB, the structure and technology classes are created in Aras. Aras structure comes with the class

Part and nested are the subclasses component, material and software shown in the figure. To avoid re-structuring the Part class (relationships, BOM, etc.) and to take advantage of Aras built-in metadata only its label is changed to Structure (only for the user view) and its subclasses are kept. Nevertheless, the subclasses Assembly, Part and Product are added in order to follow the KB product structure.

The same process is applied for the technology view. The technology class aims to describe a structure instance in more detail. What this means is that the BOM levels are still maintained within the structure view, however, the technology view can elaborate more on any structure instance through either material or manufacturing process. The test class is not added since it will be managed in Synect.

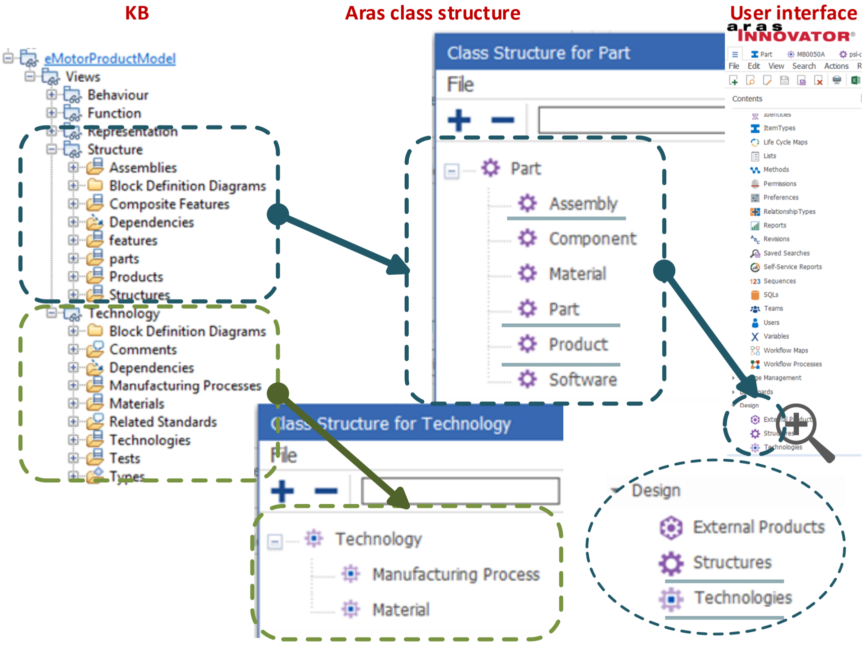
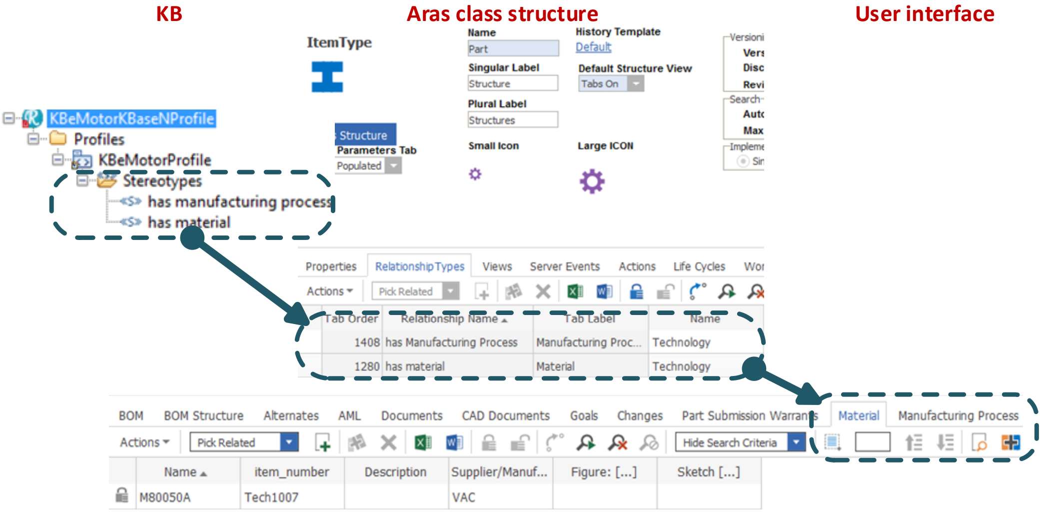


Fig. 4-15. Structure and technology class implementation.

Relationship types accomplish the aforementioned connection between structure and technology. Fig. 4-16 shows the transition from the KB association-end stereotypes to its application in PDM. The classification of the different type of structures for the electric machine is entered as a direct property on the structure class, see Fig. 4-17. The reader might ask himself why not to entered the motor structure as classes directly, well, after a couple of thoughts it is decided to enter them as a property to maintain the BOM levels within Product, Assembly and Parts and specify these motor classes instances by the selection of the property. This also avoids creating a large structure metadata that may need constant modifications. Therefore, making easier to modify a list of properties as seen in the same figure rather than a complete class metadata. Nevertheless, the user may adjust the structure to his needs. Remember that the main point of this section is to illustrate the knowledge implementation for customizing a PDM metadata.



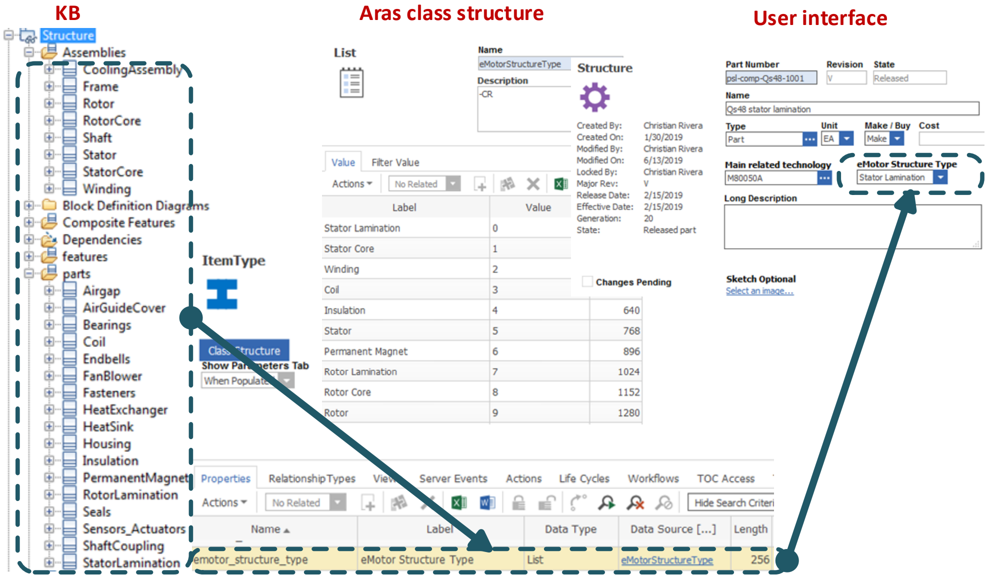
Name: Part
History Template: Default
Singular Label: Structure
Default Structure View: Tabs On
Plural Label: Structures
Parameters Tab: Populated

Tab Order	Relationship Name	Tab Label	Name
1408	has Manufacturing Process	Manufacturing Proc...	Technology
1280	has material	Material	Technology

Name	item_number	Description	Supplier/Manuf...	Figure: [...]	Sketch [...]
M80050A	Tech1007		VAC		

Fig. 4-16. Relationships implementation.

The attributes of each class are entered by extended classification. This is a flexible capability Aras provides to add groups of properties, and reference them to item-types.



Name: eMotorStructureType
Description: -CR
Show Parameters Tab: When Populated

Created By: Christian Rivera
Created On: 1/30/2019
Modified By: Christian Rivera
Modified On: 6/13/2019
Locked By: Christian Rivera
Major Rev: V
Release Date: 2/15/2019
Effective Date: 2/15/2019
Generation: 20
State: Released part

☐ Changes Pending

Label	Value	
Stator Lamination	0	
Stator Core	1	
Winding	2	
Coil	3	
Insulation	4	640
Stator	5	768
Permanent Magnet	6	896
Rotor Lamination	7	1024
Rotor Core	8	1152
Rotor	9	1280

Name	Label	Data Type	Data Source [...]	Length
emotor_structure_type	eMotor Structure Type	List	eMotorStructureType	256

Part Number: psl-comp-Qs48-1001
Revision: V
State: Released
Name: Qs48 stator lamination
Type: Part
Unit: EA
Make / Buy: Make
Cost:
Main related technology: M80050A
eMotor Structure Type: Stator Lamination
Long Description:

Fig. 4-17. Electric machine structure type.

The created structure and technology trees of this extended classification are shown in Fig. 4-18. The trees make more evident the structure followed in the KB. Finally, the attributes from the KB are entered as properties for each classification, see Fig. 4-19.

At the end of the process, a customized metadata has been set in a PDM software, permitting fast reuse and retrieval of data. In addition the standardization of the information. The aforementioned process substantiates the steps of knowledge implementation from a KB created using MOKA methodology. However, the electric motor development process also involves motor models (thermal, electromagnetic, etc.). This issue is addressed in the following subsection.

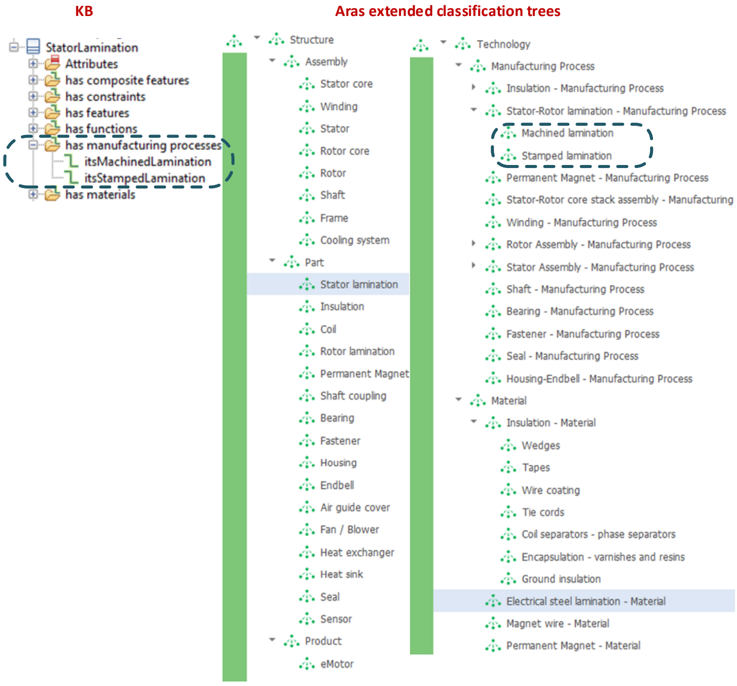
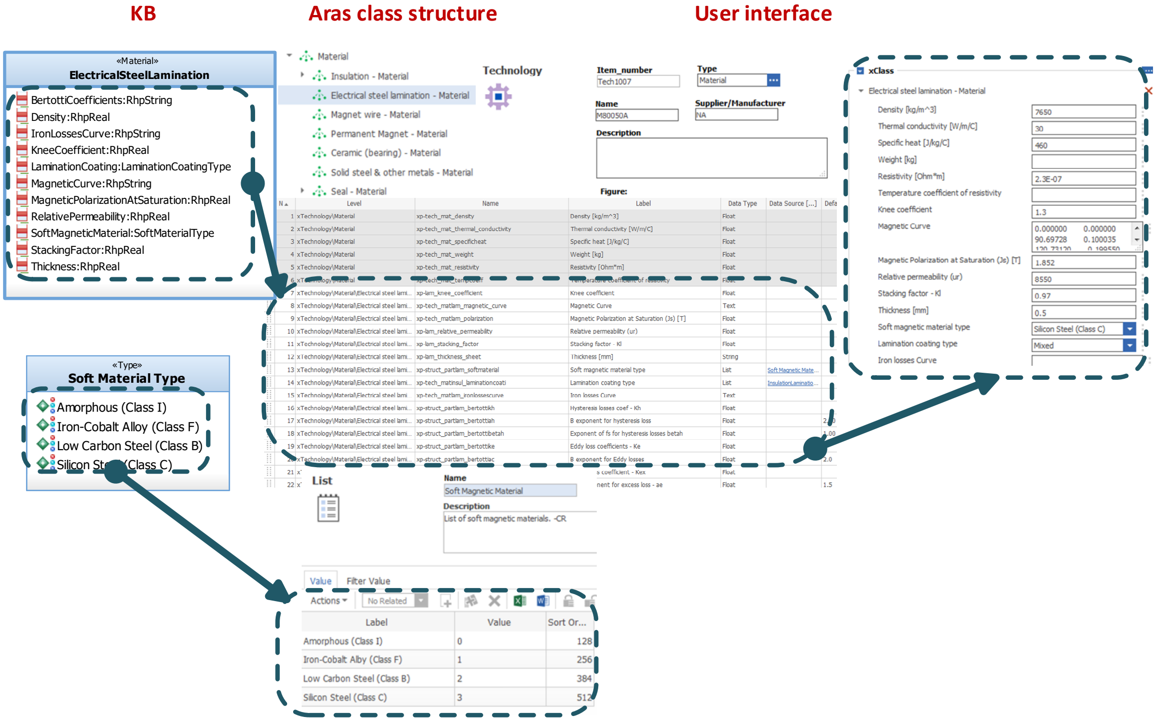


Fig. 4-18. Structure and technology view extended classification.



Item_number: Tech1007
Type: Material
Name: M80050A
Supplier/Manufacturer: NA
Description:
Figure:

Name: Soft Magnetic Material
Description: List of soft magnetic materials. -CR

Label	Value	Sort Or...
Amorphous (Class I)	0	128
Iron-Cobalt Alby (Class F)	1	256
Low Carbon Steel (Class B)	2	384
Silicon Steel (Class C)	3	512

Electrical steel lamination - Material

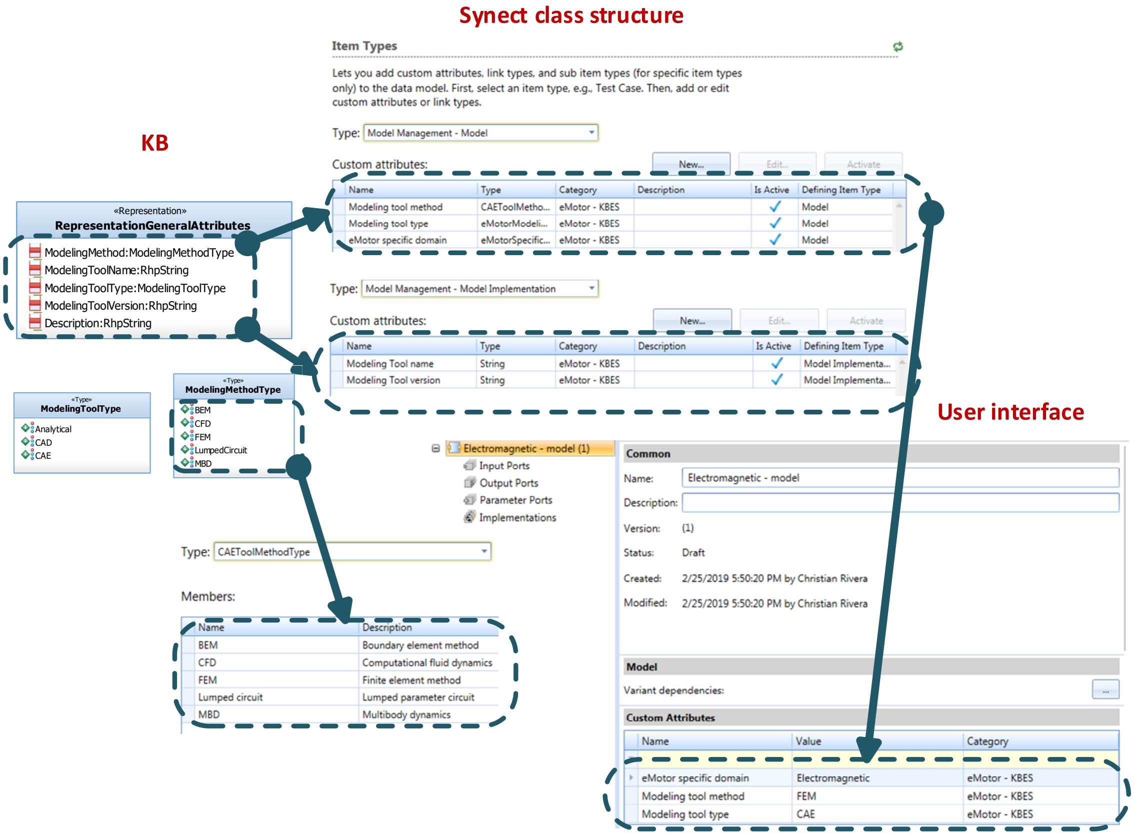
Density [kg/m^3]	7650
Thermal conductivity [W/m/C]	30
Specific heat [J/kg/C]	460
Weight [kg]	
Resistivity [Ohm*m]	2.3E-07
Temperature coefficient of resistivity	
Knee coefficient	1.3
Magnetic Curve	0.000000 0.000000 90.69728 0.100035
Magnetic Polarization at Saturation (Js) [T]	1.852
Relative permeability (ur)	8550
Stacking factor - Kl	0.97
Thickness [mm]	0.5
Soft magnetic material type	Silicon Steel (Class C)
Lamination coating type	Mixed
Iron losses Curve	

Fig. 4-19. Attributes and data types implementation.

4.6.2.2 eMachine models metadata customization

In this case Synect is used as the tool to manage the data related to motor models. These models are linked to each of the different design domains of electric machines (electromagnetic, vibro-acoustic, insulation, thermal, EMC). It is important to remark that only one PDM software can be used for the same purpose, however, to leverage the built-in metadata of Synect, this is customized to meet the purpose of this section.

In this work, the standardization of models structure and parameters is executed with the representation view. Fig. 3-20 showed the representation of a FEM tool with the properties that should be created in order to use any model regarding electromagnetic models. The chosen name of each property must be in agreement with the designers, thus, creating a standard glossary of parameters to work in a systematic approach. Fig. 4-20 shows the implementation of the general attributes into Synect metadata. As a matter of example, the modeling method attribute (see the figure) is composed of typical CAE methods implemented to model and analyze electric machines (BEM, CFD, FEM, lumped circuit, MBD). The enumeration type named CAEToolMethodType contains them; this is linked to the custom attribute modeling tool method. Finally, the user interface shows its use where FEM is selected for the electromagnetic model.



Lets you add custom attributes, link types, and sub item types (for specific item types only) to the data model. First, select an item type, e.g., Test Case. Then, add or edit custom attributes or link types.

Type: Model Management - Model

Custom attributes:

Name	Type	Category	Description	Is Active	Defining Item Type
Modeling tool method	CAEToolMetho...	eMotor - KBES		✓	Model
Modeling tool type	eMotorModeli...	eMotor - KBES		✓	Model
eMotor specific domain	eMotorSpecific...	eMotor - KBES		✓	Model

Type: Model Management - Model Implementation

Custom attributes:

Name	Type	Category	Description	Is Active	Defining Item Type
Modeling Tool name	String	eMotor - KBES		✓	Model Implementa...
Modeling Tool version	String	eMotor - KBES		✓	Model Implementa...

Type: CAEToolMethodType

Members:

Name	Description
BEM	Boundary element method
CFD	Computational fluid dynamics
FEM	Finite element method
Lumped circuit	Lumped parameter circuit
MBD	Multibody dynamics

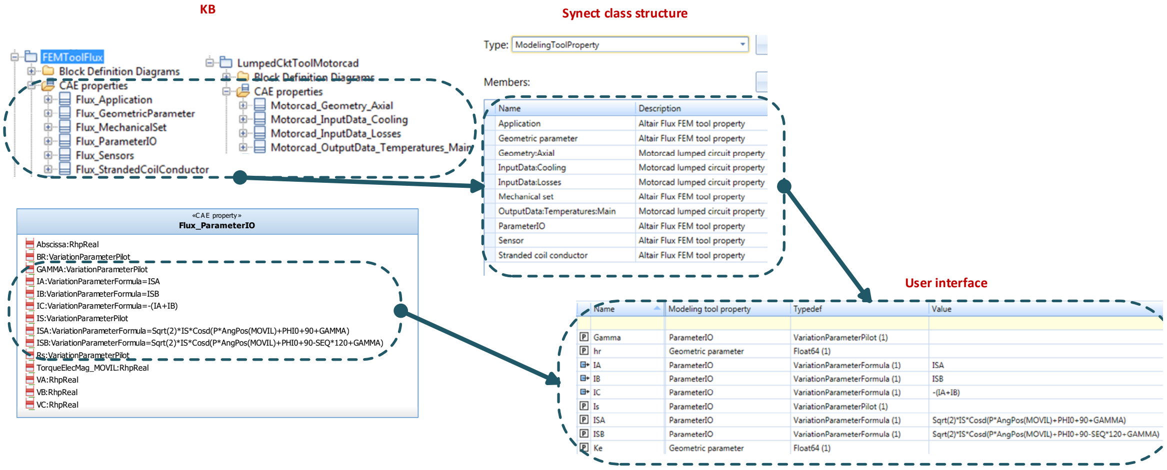
Name: Electromagnetic - model
Description:
Version: (1)
Status: Draft
Created: 2/25/2019 5:50:20 PM by Christian Rivera
Modified: 2/25/2019 5:50:20 PM by Christian Rivera

Variant dependencies:

Name	Value	Category
eMotor specific domain	Electromagnetic	eMotor - KBES
Modeling tool method	FEM	eMotor - KBES
Modeling tool type	CAE	eMotor - KBES

Fig. 4-20. General attributes for models.

Another example can be seen in Fig. 4-21. Here all the properties are implemented as a custom attribute of the item type signal-and-parameters. The parameters contained in the property Flux_ParameterIO (see Fig. 3-20) are created as parameters with their respective value (where applies). Once this is finished, the proper option among all modeling tool property can be selected. This is illustrated in the user interface of the same figure.



FEMToolFlux
- Block Definition Diagrams
- CAE properties
  - Flux_Application
  - Flux_GeometricParameter
  - Flux_MechanicalSet
  - Flux_ParameterIO
  - Flux_Sensors
  - Flux_StrandedCoilConductor

LumpedCktToolMotorcad
- Block Definition Diagrams
- CAE properties
  - Motorcad_Geometry_Axial
  - Motorcad_InputData_Cooling
  - Motorcad_InputData_Losses
  - Motorcad_OutputData_Temperatures_Main

«CAE property»
Flux_ParameterIO

Abscissa:RhpReal
BR:VariationParameterPilot
GAMMA:VariationParameterPilot
IA:VariationParameterFormula=ISA
IB:VariationParameterFormula=ISB
IC:VariationParameterFormula=-(IA+IB)
IS:VariationParameterPilot
ISA:VariationParameterFormula=Sqrt(2)*IS*Cosd(P*AngPos(MOVIL)+PHI0+90+GAMMA)
ISB:VariationParameterFormula=Sqrt(2)*IS*Cosd(P*AngPos(MOVIL)+PHI0+90-SEQ*120+GAMMA)
Rs:VariationParameterPilot
TorqueElecMag_MOVIL:RhpReal
VA:RhpReal
VB:RhpReal
VC:RhpReal

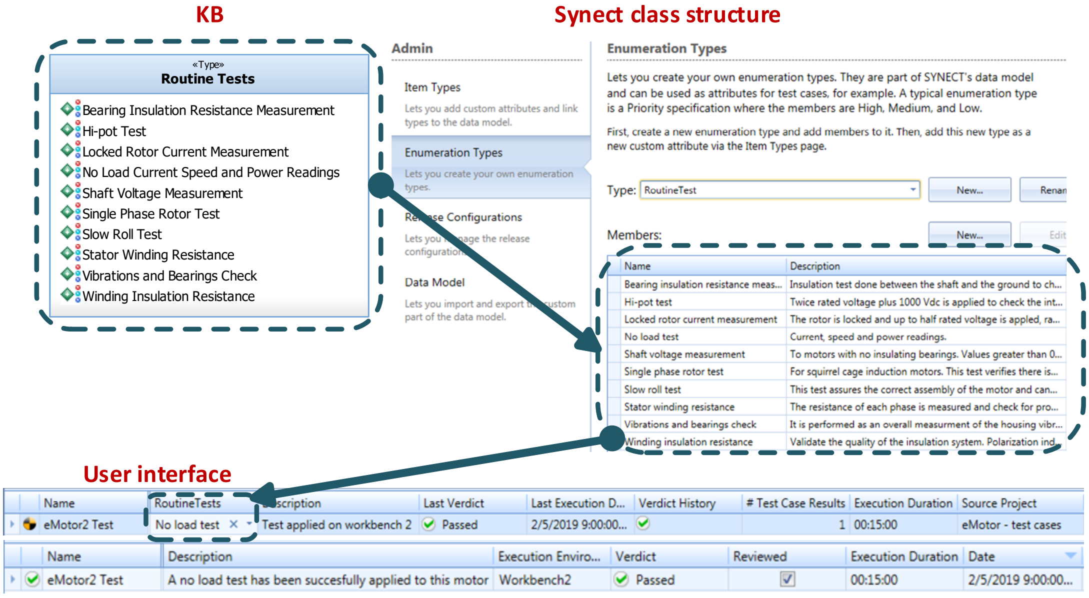
Type: ModelingToolProperty

Name	Description
Application	Altair Flux FEM tool property
Geometric parameter	Altair Flux FEM tool property
Geometry:Axial	Motorcad lumped circuit property
InputData:Cooling	Motorcad lumped circuit property
InputData:Losses	Motorcad lumped circuit property
Mechanical set	Altair Flux FEM tool property
OutputData:Temperatures:Main	Motorcad lumped circuit property
ParameterIO	Altair Flux FEM tool property
Sensor	Altair Flux FEM tool property
Stranded coil conductor	Altair Flux FEM tool property

Name	Modeling tool property	Typedef	Value
Gamma	ParameterIO	VariationParameterPilot (1)	
hr	Geometric parameter	Float64 (1)	
IA	ParameterIO	VariationParameterFormula (1)	ISA
IB	ParameterIO	VariationParameterFormula (1)	ISB
IC	ParameterIO	VariationParameterFormula (1)	-(IA+IB)
Is	ParameterIO	VariationParameterPilot (1)	
ISA	ParameterIO	VariationParameterFormula (1)	Sqrt(2)*IS*Cosd(P*AngPos(MOVIL)+PHI0+90+GAMMA)
ISB	ParameterIO	VariationParameterFormula (1)	Sqrt(2)*IS*Cosd(P*AngPos(MOVIL)+PHI0+90-SEQ*120+GAMMA)
Ke	Geometric parameter	Float64 (1)	

Fig. 4-21. Signal and parameters representation view implementation.

The same process is followed for the tests shown in Fig. 3-18b. Enumeration types of the different tests cases are created for the user to select the proper one among them. Fig. 4-22 presents this transition from the KB to the user interface.



«Type»
Routine Tests

Bearing Insulation Resistance Measurement
Hi-pot Test
Locked Rotor Current Measurement
No Load Current Speed and Power Readings
Shaft Voltage Measurement
Single Phase Rotor Test
Slow Roll Test
Stator Winding Resistance
Vibrations and Bearings Check
Winding Insulation Resistance

Item Types: Lets you add custom attributes and link types to the data model.

Enumeration Types: Lets you create your own enumeration types.

Data Model: Lets you import and export custom part of the data model.

Enumeration Types

Lets you create your own enumeration types. They are part of SYNECT's data model and can be used as attributes for test cases, for example. A typical enumeration type is a Priority specification where the members are High, Medium, and Low.

First, create a new enumeration type and add members to it. Then, add this new type as a new custom attribute via the Item Types page.

Type: RoutineTest

Members:

Name	Description
Bearing insulation resistance meas...	Insulation test done between the shaft and the ground to ch...
Hi-pot test	Twice rated voltage plus 1000 Vdc is applied to check the int...
Locked rotor current measurement	The rotor is locked and up to half rated voltage is appled, ra...
No load test	Current, speed and power readings.
Shaft voltage measurement	To motors with no insulating bearings. Values greater than 0...
Single phase rotor test	For squirrel cage induction motors. This test verifies there is...
Slow roll test	This test assures the correct assembly of the motor and can...
Stator winding resistance	The resistance of each phase is measured and check for pro...
Vibrations and bearings check	It is performed as an overall measurment of the housing vibr...
Winding insulation resistance	Validate the quality of the insulation system. Polarization ind...

Name	RoutineTests	Description	Last Verdict	Last Execution D...	Verdict History	# Test Case Results	Execution Duration	Source Project
eMotor2 Test	No load test	Test applied on workbench 2	✓ Passed	2/5/2019 9:00:00...	✓	1	00:15:00	eMotor - test cases

Name	Description	Execution Enviro...	Verdict	Reviewed	Execution Duration	Date
✓ eMotor2 Test	A no load test has been succesfully applied to this motor	Workbench2	✓ Passed	☑	00:15:00	2/5/2019 9:00:00...

Fig. 4-22. Tests for eMotors, technology view implementation.

At the end of this implementation a standard template has been stablished for the user to use on every new project. The template provides the customized metadata, parameters and units avoiding the designer to spend additional time on their creation and at the same time standardizing the process. Fig. 4-23 shows the resulting template.

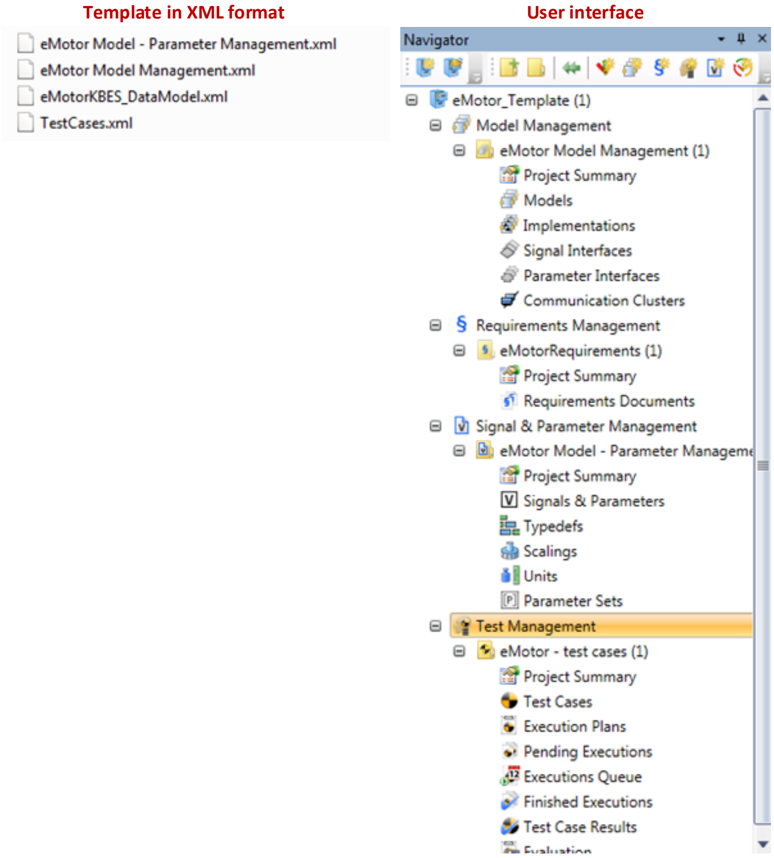


Fig. 4-23. eMachine template in Synect for models data management.

4.7 Discussion

The KBE system framework is encapsulated in the KBV2-model. The model provides a holistic view of the process flow to create KBE applications and integrate them into the eMachine development process. The KBV2-model is composed of the Knowledge Base generation process and the eMachine standardized macro-level framework defined to show the purpose of the KBE application into the system of the electric machine manufacturer. The actors were identified as well as their role in the proposed framework. A list of tools were provided to generalize the implementation of the KBE system framework. In addition, the knowledge management and the KBE system architecture were described for a complete and well structure development of a KBE system.

The framework is the result of a thorough analysis of several KBE project implementations, the proposed underlying structure is considered by the author the most appropriate to guide the actors to achieve the expected results from a KBE application. However, it has the limitations of not including a managerial perspective that is important to analyze costs and risks of a KBE solution in an enterprise. It also lacks guidelines on communication channels among actors and lacks a software development methodology for the KBE application and commercial off the shelf software (COTS) integration. Nevertheless, the processes were mentioned throughout the KBES framework description and the user of this framework can include a solution for each case.

To provide examples about the use of the KBE system framework, implementations are made for the development of micro and macro level KBE applications in the following chapters.

Chapter 5

MICRO-LEVEL IMPLEMENTATION

The purpose of this chapter is to provide substantiation of the proposed KBES framework for the micro-level processes. Two KBE applications are developed, one to compute brushless motor operating points and the other to query performance characteristics data directly from a FEM tool.

Section 5.1 introduces the chapter. Section 5.2 presents the first micro-level KBE application called eMachine Operating Limits, section 5.3 shows the second KBE application called eMotor Performance Characteristics Finder. Finally, section 5.4 discusses the results.

5.1 Introduction

The previous chapter addressed the eMachine product model implementation with one example in PDM metadata customization. Nevertheless, the eMachine process metamodel has not yet been implemented in detail. This chapter presents implementation of the KBE system framework regarding the micro-process metamodel providing substantiation of the process.

Moreover, two micro-level KBE applications are developed. The second application integrates the first one. The first application has the capability of computing motor operating points through new algorithms coupled to FEM, targeting complex operating points for brushless AC motors. A comparison study is carried out by computing the MTPA operating point with and without the KBE application.

The second application expands the capabilities of the first one. The developed micro-level KBE application supports designers to query the performance data of a given operating point of a brushless motor using the electromagnetic and thermal domains. What makes challenging this task is that the performance data is obtained directly from a finite element analysis tool searching the point on one of the feasible control regions MTPA, FW and MTPV. In other words, the user has not to prepare a sweep of simulations by himself, and has not to post-process the simulation results to get the performance data.

Moreover, the searching algorithm founds the queried point on one of the feasible regions i.e. the best possible performance of the motor is always obtained. The process is focus in the formalize step in order to allow replicability. Finally, a use case example making a comparison study of performance between a ferrite-assisted synchronous reluctance motor and synchronous reluctance motor is carried out.

5.2 KBE application: eMachine operating limits

Fig. 5-1 shows the process to develop the devised KBE application. In Fig. 5-1a the process to develop the respective knowledge base is shown, the example tries to illustrate the liberty to create schemes to clarify the process for the actors. Likewise, Fig. 5-1b shows a general scheme of what the KBE application is capable of doing. To elaborate more, the figure shows that the end-user can call the proper function to compute a desired operating point from the KBE application. It also shows that the KBE application replies the end-user with acceptance or warnings of errors and parameters out of range (not specified in the figure). In addition, the KBE application needs an electromagnetic model as input to operate and as outputs; it returns the characteristic variables seen in the figure.

Moreover, it can be seen that related to the V2-model the KBE application can be applied in the domain-specific design stage for any of the three V-cycles.

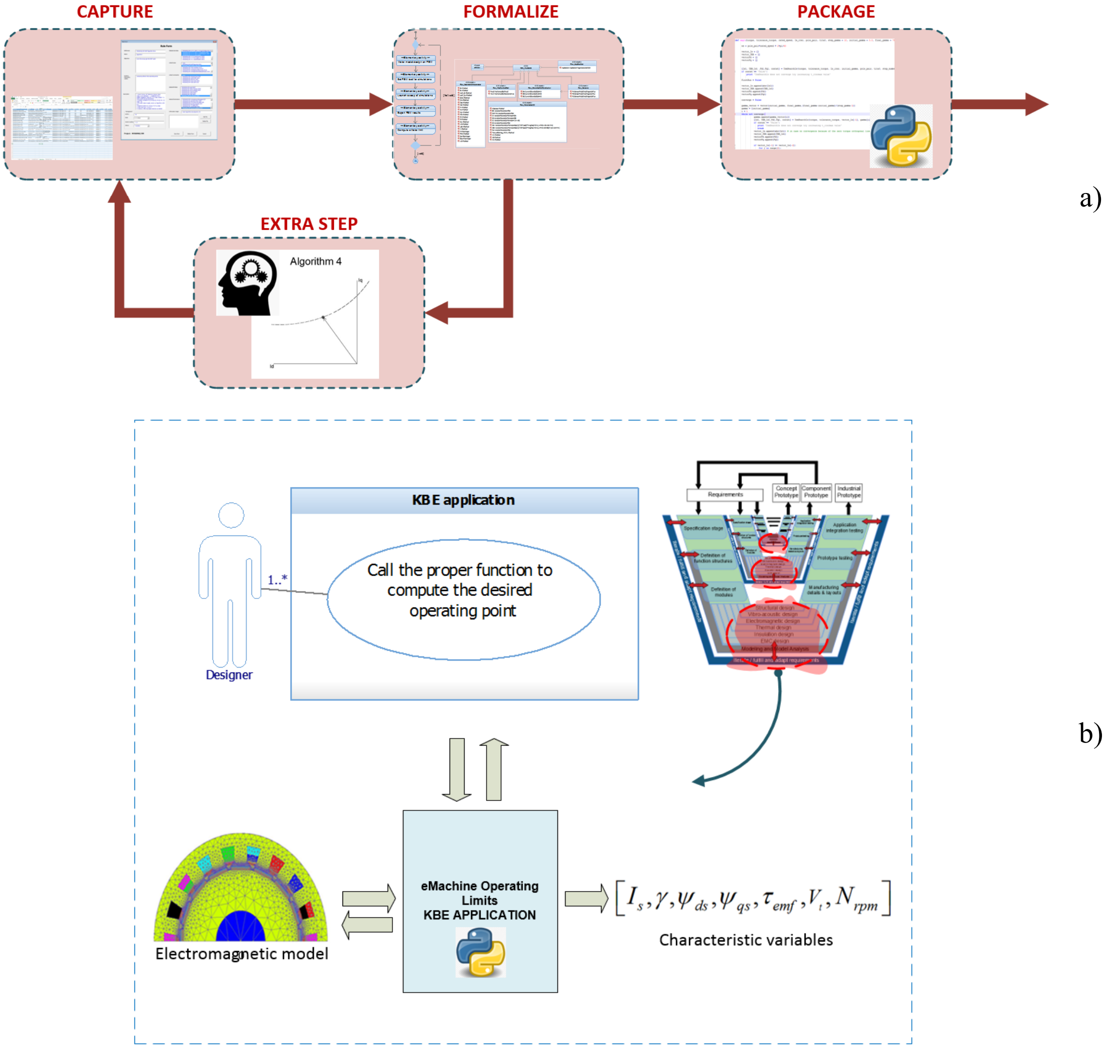


Fig. 5-1. Model to develop the KBE application, eMachine operating limits; a) Knowledge Base Generation with knowledge feedback; b) General scheme of the KBE application and its usage in the V2-model.

The KBE application and its development process is now explained in more detail.

5.2.1 Identify and justify: problem context

The problem context is now described. In the conceptual phase of the design, where changes to features of the model such as geometry is repetitive in order to find a valid model that can later be completely characterized. Fig. 5-2a shows a typical process followed to compute a motor's characteristics using FEM. Once the model is set and ready, a sweep of simulation points are done by introducing a range of three sinusoidal phase currents, the outputs are (but not limited to) fluxes, currents and electromagnetic torque. After the results are ready, the designer exports them for post-processing in order to obtain the characteristics that satisfies requirements and study the performance of the machine at different load points.

In general and regarding electrical machine design, FEM tools compute a working point targeted with phase currents, however designers are also interested in results that cannot be targeted directly such as:

1. Given the torque and angle, get the current magnitude that produces that torque.
2. Given the speed and voltage reference and a given d axis current, get the current vector where that voltage is reached.
3. Get the characteristic current (I_{ch}) of the electric motor.
4. Given the torque what is the MAPT point that produces that torque.
5. Given the current limit what is MTPA point for that current.
6. Given the current and voltage limit get the FW speeds along the current circle.
7. Given a speed and voltage limit what is the MTPV on the voltage ellipse.
8. Given a voltage limit what is the mode 1 limit (MTPA-FW boundary) see algorithm 8 in Fig. 5-3
9. Given the voltage and current limit get the mode 2 limit (FW-MTPV boundary) see algorithm 9 in Fig. 5-3, the CPSR point and the maximum power to speed ratio (MPSR) point of the electric motor.

The goal is to accomplish the **automatic computation** of the eMotor limits targeting the 9 points previously listed. In this way, reducing the workload of the designer and supporting his design process. Fig. 5-2 shows the reduction of steps that wants to be achieved by developing this KBE application. It is concluded that the reduction of three semi-automatic processes in one full automatic process will save significant workload and time to a designer. At the same time, the evaluated efforts to automate this process are not costly.

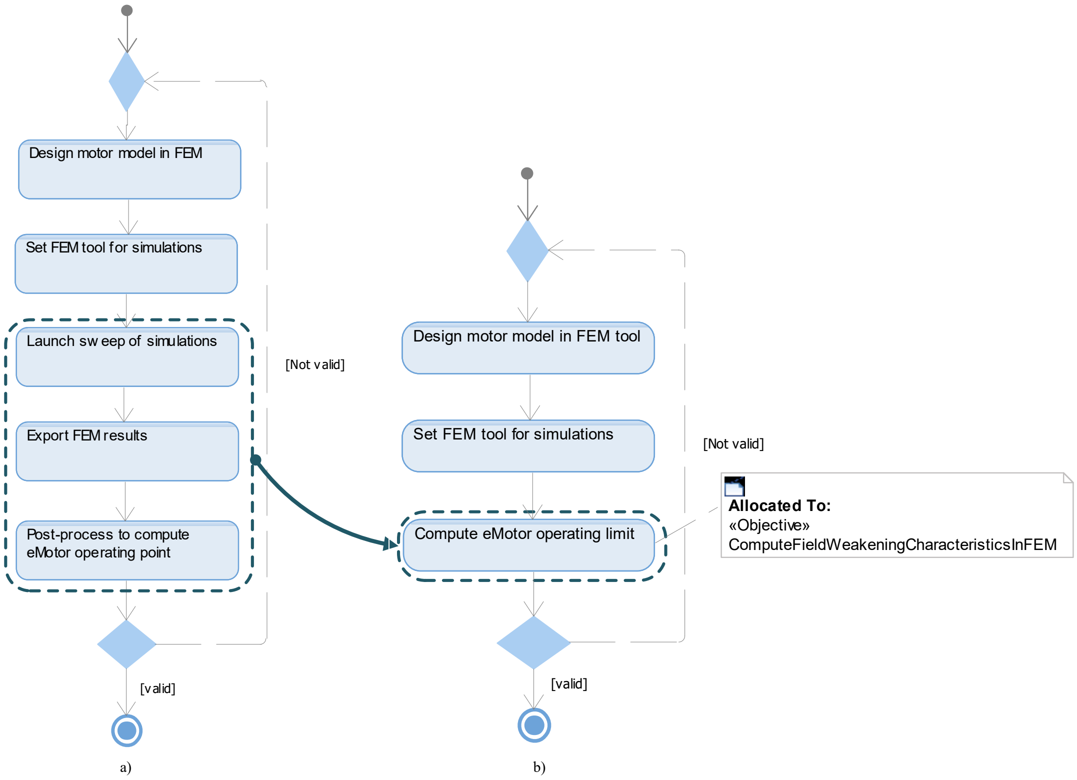


Fig. 5-2. a) Typical process for motor limits computation using FEM. b) Process using the KBE application.

5.2.2 Capture and formalize

Retaking the Knowledge Base generation process the knowledge has been captured for each activity of Fig. 5-2a. The first two activities do not change. However, previously a standardized way of representation for the FEM tool has been shown in Fig. 3-20. This allows the user to set the FEM tool following the same glossary of terms as any other fellow designer.

A step further is taken outside MOKA methodology with the creation of intelligent algorithms that support the designer with the computation of these operating points. In this way, during the automation process new knowledge was generated enriching the capture stage. Fig. 5-1a shows how in many cases a new knowledge feedback can be added during the process.

Nine algorithms have been created to accomplish the computation of the operating points previously listed. The algorithms take the advantage of the high fidelity non-linear electromagnetic environment of FEM tools to compute operation points such as MTPA, FW, MTPV and others that cannot be targeted directly, therefore, considering variable dependencies that are difficult to model analytically with high precision. Fig. 5-3 illustrates how each algorithm works and what operating points it can compute on the circle diagram.

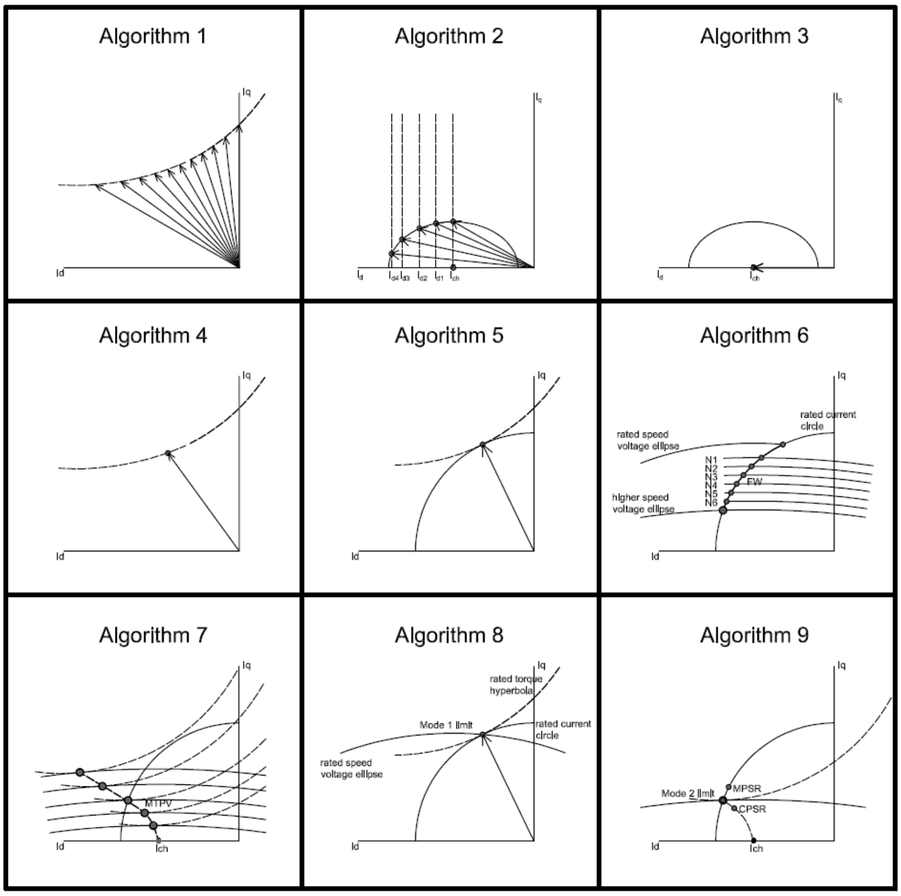
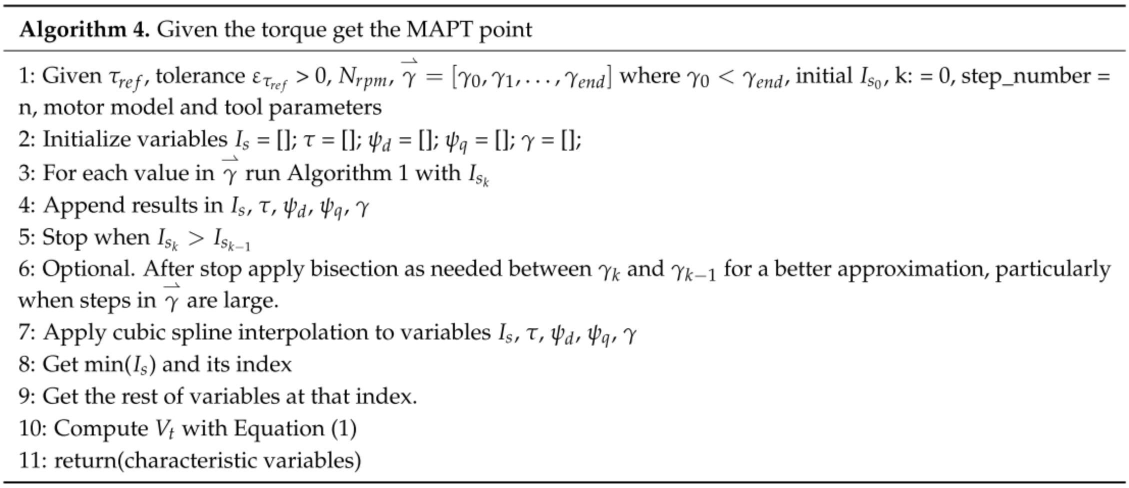


Fig. 5-3. Algorithms' target points on the motor circle diagram.

To shorten this section let us focus on algorithm 4 which targets the value of MAPT point for a given torque, in summary what is done is to compute the constant torque curve for different angles using Algorithm 1, then using cubic spline interpolation obtain the minimum current magnitude. The MAPT operating point is the same MTPA point but targeted with the torque instead of the current. The reader is encourage to read Appendix B where a complete description is given for each algorithm as well as their validation.



Algorithm 4. Given the torque get the MAPT point

```
Given $\tau_{ref}$, tolerance $\varepsilon_{\tau_{ref}} > 0$, $N_{rpm}$, $\overrightarrow{\gamma} = [\gamma_0, \gamma_1, \ldots, \gamma_{end}]$ where $\gamma_0 < \gamma_{end}$, initial $I_{s_0}$, k: = 0, step_number = n, motor model and tool parameters
Initialize variables $I_s$ = []; $\tau$ = []; $\psi_d$ = []; $\psi_q$ = []; $\gamma$ = [];
For each value in $\overrightarrow{\gamma}$ run Algorithm 1 with $I_{s_k}$
Append results in $I_s$, $\tau$, $\psi_d$, $\psi_q$, $\gamma$
Stop when $I_{s_k} > I_{s_{k-1}}$
Optional. After stop apply bisection as needed between $\gamma_k$ and $\gamma_{k-1}$ for a better approximation, particularly when steps in $\overrightarrow{\gamma}$ are large.
Apply cubic spline interpolation to variables $I_s$, $\tau$, $\psi_d$, $\psi_q$, $\gamma$
Get min($I_s$) and its index
Get the rest of variables at that index.
Compute $V_t$ with Equation (1)
return(characteristic variables)
```

Next, the algorithms become part of the rules in the knowledge book. Following the new micro-process knowledge representation presented in Section 3.4.4. The algorithms are formalized as indications cnst within each specific objective. The knowledge model can be seen in Fig. 3-26. The process to use the KBE application in FEM is the one shown in Fig. 5-2b. The figure shows the indication action "Compute eMotor operating limit" is allocated to a higher level objective called "Compute Field Weakening Characteristics in FEM". From this objective derives each of the objectives in Fig. 3-26 (see Fig. 5-4). It is important to remark that the knowledge models are kept within the same knowledge modeling tool (IBM rational Rhapsody for this case), before a link to the rule had to be included in order to seek it in the knowledge book, see Fig. 3-22.

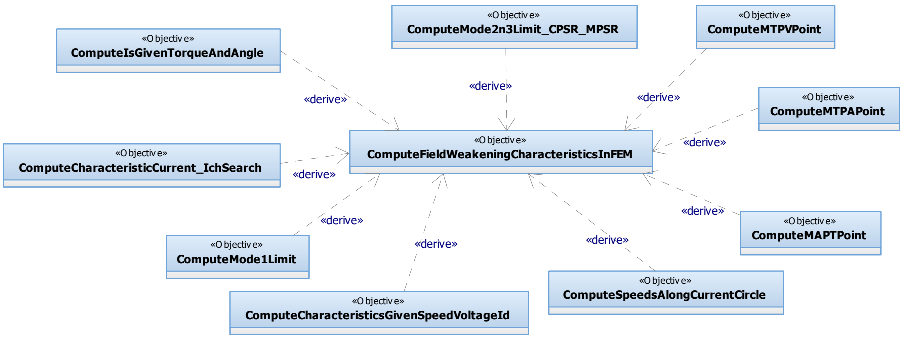


Fig. 5-4. Derivation of objectives for the KBE application.

5.2.3 Package

The package stage of MOKA is carried out by implementing the algorithms in Jython, this is the language coupled to Flux for the automation of processes in FEM. A module is developed permitting the user to import it into a script.

The activate stage is shown in the following subsection. Which makes a computational time comparison between the processes presented in Fig. 5-2.

5.2.4 KBE application use case example: Compute MTPA (comparison of process with and without the KBE application)

To compare the computation time consumption in both processes with similar conditions an example is now given with a ferrite IPM motor. For comparison only computation time is considered, not any human work time to develop a task is taken into account, although it is evident the conventional process involves more work. Suppose we have just finished a new IPM motor geometry design with the features mentioned in Table 5-1 looking to satisfy the rated values specified in Table 5-2. The best current phasor satisfying these requirements is certainly on the MTPA curve. However, we do not know what that current phasor is. So with no prior analysis the current phasor will be found using both processes mentioned in Fig. 5-2. The application is set to magnetostatic 2D, the tool use is Flux 2D/3D from Altair, it has Jython as the coupled language.

The process “set FEM tool for simulations” only refers to prepare the model to execute simulations like faces, regions, mesh, physics, materials, etc. Because this is common for both processes the time in the comparison is not considered.

Table 5-1 IPM motor parameters

Parameter	Value
Number of slots	54
Pole pairs	3
Length	175 mm

Table 5-2. Motor requirements

Parameter	Value	Units
Torque	370	Nm
Current	≤12	A
Voltage	380	V
Length	<180	mm

The sweep of simulations for the conventional process is launched with the following range of values, assuming the result lies between these values:

1. Range to search the MTPA current angle: [5°, 65°] in 10 steps.
2. Currents magnitude range for each angle: [8 A, 15 A] in 8 steps.
3. Number of steps in one-sixth of the electrical period (position angle): 3.

The time to finish the simulations was of 1457.37 s. The results were exported to excel, to be used in Matlab for post-processing. The time of exporting the results was of 91.02 s. Applying pre-developed post-processed scripts after the results are imported in Matlab the MTPA for each current is found as it is shown in Fig. 5-5. The computation time to find the current on the MTPA curve that satisfies the rated torque value was of 0.4315 s.

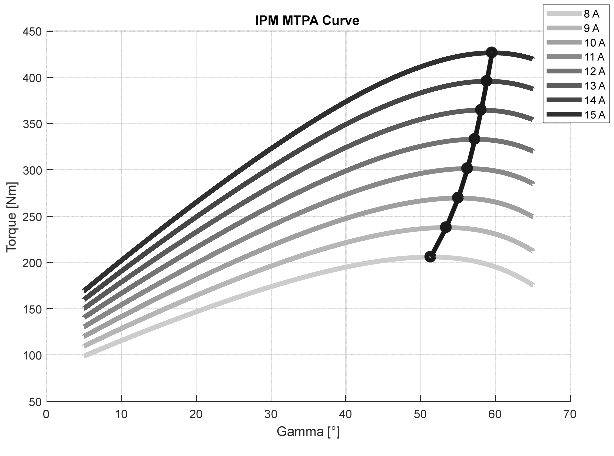


Fig. 5-5. Post-processed MTPA curve.

Now, for the process using the algorithms. Algorithm 4 computes the minimum current vector for a given torque value. For the reference torque of 370 Nm the algorithm inputs were set as follows:

1. Torque tolerance 1×10^{-2} Nm
2. Current angle range [5°, 65°] in 10 steps.

3. Current magnitude initial guess 10 A
4. Number of steps in one-sixth of the electrical period (position angle): 3.

The process lasted a total of 1076.83 s. The current results for each process and the computation time consumption comparison are summarized in Table 5-3. The computation time results suggest that in similar conditions using the algorithms will give the advantage to the designer of reducing computational time to obtain a characteristic, besides less work to do it. The results show the current magnitude is above requirements then changes to the model must be made in order to satisfy them and the process should be repeated. The developed KBE application provide the user the possibility to generate different machine candidates within the same FEM tool taking into account all multi-physics effects of FEM. This example was given with Algorithm 4 to compute the minimum current phasor for a given torque, which is one of the algorithms that takes longer to give a result, nevertheless, Algorithms 1, 2, 3 and 6 are algorithms that usually take less than 5 min to provide an answer. The use and selection of the algorithms must be adjusted by the designer needs to get the most out of it.

Table 5-3. Computation time comparison for both processes.

Process	Typical Process Time	Process Using the Algorithms
Launch sweep of simulations	1457.37 s	0.00 s
Export FEM results	91.02 s	0.00 s
Compute MTPA current phasor	0.4315 s	1076.83 s
Total time	1548.82 s	1076.83 s
Current result	$13.1746 \angle 58.23°$	$13.35 \angle 58.40°$

In conclusion, the computational time for one of the worst-case scenarios of application is 30% less than the normal process followed by the designer. Being able to optimize its use even more and not considering the human work involved. The KBE application successfully accomplished the goal of supporting the designer to compute complex points reducing computational time and workload.

5.3 KBE application: eMotor performance characteristics finder

To highlight the MVP principle in some extent for micro-level development. This KBE application incorporates and expands the capabilities of the previous application regarding the design of brushless electric motors for the micro-level case. To avoid clutter and based on the contribution in knowledge formalization of this work, the implementation of the framework is presented with more emphasis in the "formalize" the knowledge stage. Fig. 5-6 shows the KBV2-model for the KBE application. The model specifies where the KBE application will be deployed. It can be seen that it can be used for the domain-specific design phase of any of the V-cycles involving the electromagnetic and thermal domains. The model also provides a scheme showing in general form the capabilities of the application as well as the tools for its implementation. This is explained in more detail in the problem context.

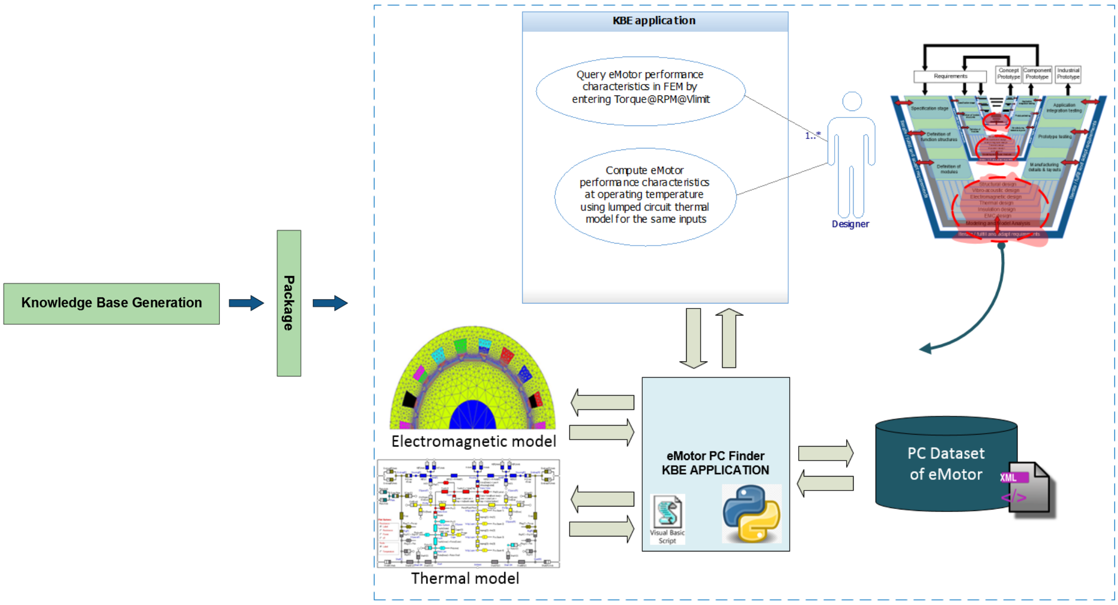


Fig. 5-6. KBV2-model for the KBE application: eMotor PC finder.

5.3.1 Identify and justify: problem context

eMotor designers need to make accurate performance computations during the design stage of AC brushless electric motors. Through the design process, they compute the performance characteristics (currents, voltages, torque, losses, power, etc.) for different operating points to analyze the behavior of the eMotor. For complex models were analytical models are not available, the best way to obtain these results is through FEM simulations. The current approach followed by the designers is, first, to make a sweep of simulations in FEM, second, to post-process the results, then, to analyze the data. Finally, if the designer considers making changes to the electromagnetic model of the eMotor, then, the process (if required) should be restarted. They also experienced the disadvantage of making too many unnecessary simulations. Additionally, they also require computing for the same operating point the thermal performance of the motor. Therefore, the KBE application shall query and retrieve the performance data of a requested operating point for the user to analyze. The process should be as fast as possible and with the least effort by the designer based on the inputs: torque, speed, and voltage limit.

The capabilities agreed are that the KBE application shall obtain automatically the performance data for the requested point of operation entered in the form of TORQUE@SPEED@Vlimit. The point shall be found in the feasible optimal operating region, this region is one of the three type of controls MTPA, FW, and MTPV limited by the converter voltage; see Fig. 5-7. If not found in any, the data should be saved but the user shall be notified the requested point exceeds the voltage limit showing the maximum achievable point. Every result shall be automatically saved for further use. The KBE application should be able to retrieve the requested point if it has already been computed (never repeat the same simulation). In this manner, saving time and making the computational process more efficient and intelligent. Finally, it shall allow the designer to have the optional capability of computing the performance data in operating temperature conditions by computing the change in temperature by a lumped parameter circuit model and changing automatically the parameters in the FEM model for performance data computation in one of the three modes of operation.

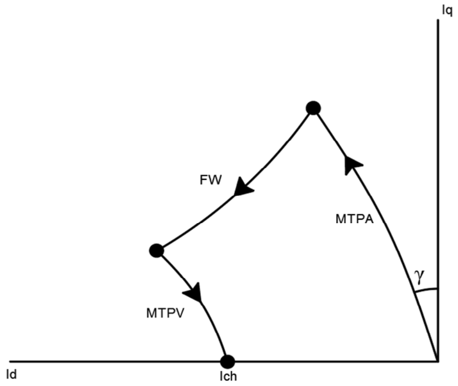


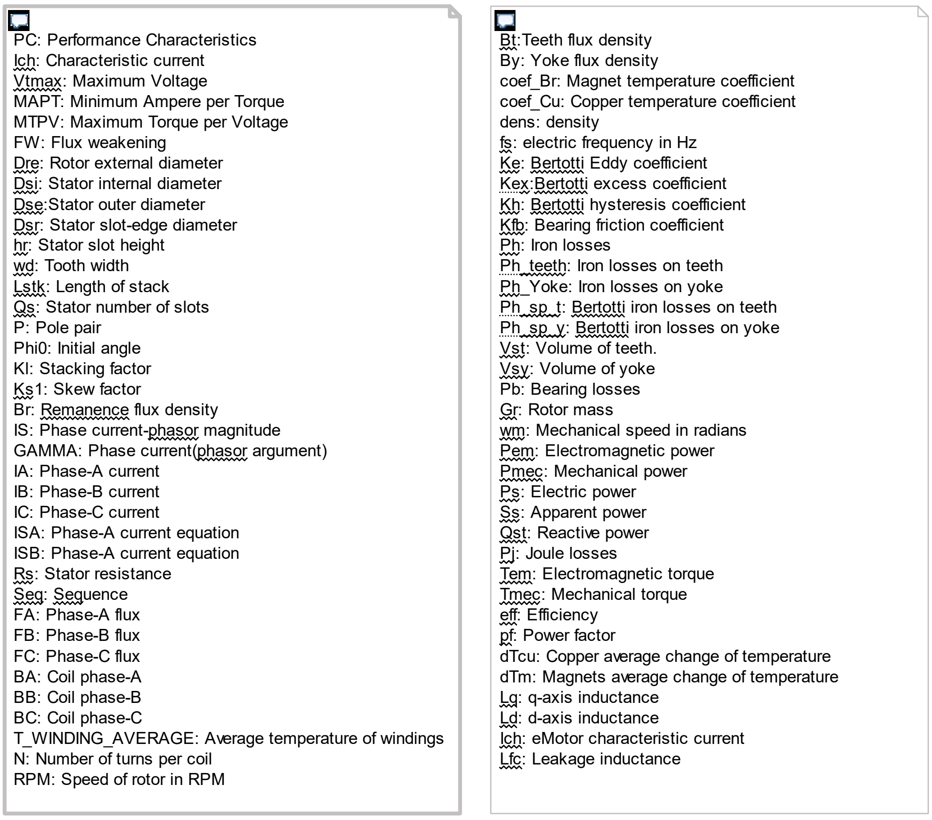
Fig. 5-7. Feasible regions for searching of operating point.

The design tools used by the designers are Altair Flux 2D and Motorcad so these are mandatory for the KBE application. Regarding the knowledge-modeling tool, IBM Rational Rhapsody 8.3.1 is selected and for coding Jython and VBScript as programming language since the former is the language embedded for Flux and the latter for Motorcad. The results are presented in an XML structure format for the user to read it as XML or to visualize it mainly through Microsoft Excel.

5.3.2 Capture and formalize

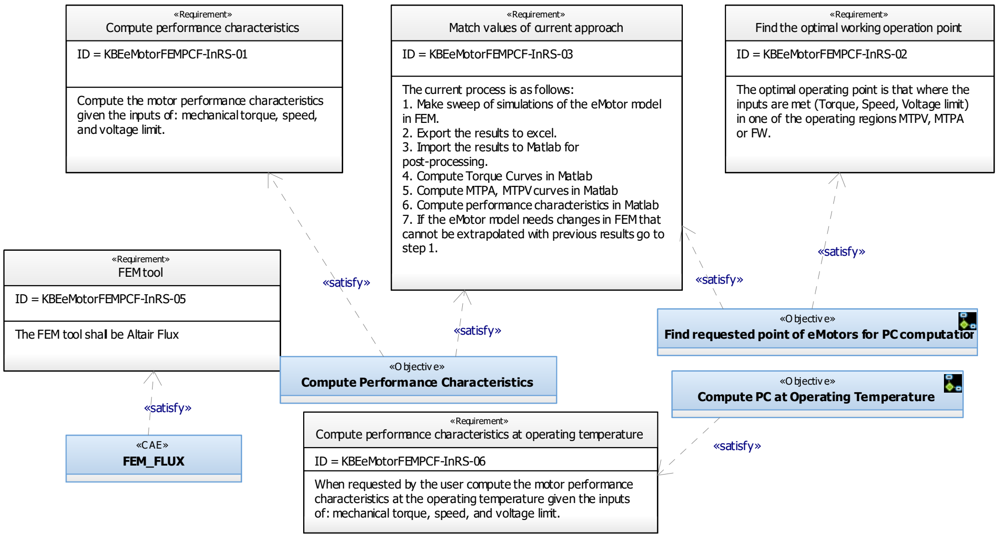
As with the previous example, the knowledge for this application was gathered from general eMachine theory and validated through experts. This subsection explains how the problem is solved through the formalized knowledge. The KBE project named eMotorPCFinder is organized in IBM Rational Rhapsody as illustrated in Fig. 4-12. From the figure, it can be seen that in the first level we have the specific project KBE unit called eMotorPCFinder as well. Followed by the profiles and finally the generic KB called KBeMotor_v1. As it can be seen in the same figure, the KBE project unit is divided into the process model, product model, and SysML refinement.

The SysML refinement package contains diagrams that elaborate more about the KBE application for better understanding. Please recall the term package here is related to SysML terms not to the MOKA package step. It was created as a separate package to organize all SysML diagrams outside the KBeMotorProfile. In this way, to illustrate and provide an example to the reader, however, SysML diagrams can be created and organized within any other packages (product or process) in the project. Fig. 5-8 shows the glossary contained within the SysML refinement package with the variables used throughout the project.


PC: Performance Characteristics
Ich: Characteristic current
Vtmax: Maximum Voltage
MAPT: Minimum Ampere per Torque
MTPV: Maximum Torque per Voltage
FW: Flux weakening
Dre: Rotor external diameter
Dsi: Stator internal diameter
Dse:Stator outer diameter
Dsr: Stator slot-edge diameter
hr: Stator slot height
wd: Tooth width
Lstk: Length of stack
Qs: Stator number of slots
P: Pole pair
Phi0: Initial angle
Kl: Stacking factor
Ks1: Skew factor
Br: Remanence flux density
IS: Phase current-phasor magnitude
GAMMA: Phase current(phasor argument)
IA: Phase-A current
IB: Phase-B current
IC: Phase-C current
ISA: Phase-A current equation
ISB: Phase-A current equation
Rs: Stator resistance
Seq: Sequence
FA: Phase-A flux
FB: Phase-B flux
FC: Phase-C flux
BA: Coil phase-A
BB: Coil phase-B
BC: Coil phase-C
T_WINDING_AVERAGE: Average temperature of windings
N: Number of turns per coil
RPM: Speed of rotor in RPM

Bt:Teeth flux density
By: Yoke flux density
coef_Br: Magnet temperature coefficient
coef_Cu: Copper temperature coefficient
dens: density
fs: electric frequency in Hz
Ke: Bertotti Eddy coefficient
Kex:Bertotti excess coefficient
Kh: Bertotti hysteresis coefficient
Kfb: Bearing friction coefficient
Ph: Iron losses
Ph_teeth: Iron losses on teeth
Ph_Yoke: Iron losses on yoke
Ph_sp_t: Bertotti iron losses on teeth
Ph_sp_y: Bertotti iron losses on yoke
Vst: Volume of teeth.
Vsy: Volume of yoke
Pb: Bearing losses
Gr: Rotor mass
wm: Mechanical speed in radians
Pem: Electromagnetic power
Pmec: Mechanical power
Ps: Electric power
Ss: Apparent power
Qst: Reactive power
Pj: Joule losses
Tem: Electromagnetic torque
Tmec: Mechanical torque
eff: Efficiency
pf: Power factor
dTcu: Copper average change of temperature
dTm: Magnets average change of temperature
Lq: q-axis inductance
Ld: d-axis inductance
Ich: eMotor characteristic current
Lfc: Leakage inductance

Fig. 5-8. Glossary of variables for eMotorPCFinder.



Compute the motor performance characteristics given the inputs of: mechanical torque, speed, and voltage limit.

The current process is as follows:
1. Make sweep of simulations of the eMotor model in FEM.
2. Export the results to excel.
3. Import the results to Matlab for post-processing.
4. Compute Torque Curves in Matlab
5. Compute MTPA, MTPV curves in Matlab
6. Compute performance characteristics in Matlab
7. If the eMotor model needs changes in FEM that cannot be extrapolated with previous results go to step 1.

The optimal operating point is that where the inputs are met (Torque, Speed, Voltage limit) in one of the operating regions MTPV, MTPA or FW.

The FEM tool shall be Altair Flux

When requested by the user compute the motor performance characteristics at the operating temperature given the inputs of: mechanical torque, speed, and voltage limit.

Fig. 5-9. Requirement diagram of the micro-level KBE application.

Moreover, the package also withholds the requirement diagrams that support the actors (for this stage in particular the knowledge engineer) to ensure the satisfaction and traceability of requirements of the KBE application. Fig. 5-9 shows an example with some of the requirements and objectives. Finally, the SysML refinement package also holds the use case diagram shown in the scheme of Fig. 5-6 and contains diagrams that supports the SW development process (package step).

Now let us start with the product model. In this case, the KBE application will not have as outcome a complete design of a product, part or assembly. Instead, it will support the designer with complex multiphysics computation for the performance analysis of the design. Fig. 5-10 illustrates the function diagram of the eMotor (the product) regarding the KBE application context. The eMotor works within the feasible optimal operating region that is executed through the control strategies MTPA, FW and MTPV. For this KBE project, the algorithms developed in the previous applications will be reused. See the package Field Weakening FEM Coupled Algorithms in Fig. 4-12 and the algorithms knowledge representation in Fig. 3-26. Therefore, as technical solutions for the analysis we have the algorithms shown in the function diagram. For the computation of the performance within requirements, the tools Motorcad and Flux are used and presented in the diagram. In addition, the product class “eMotor” is reused from the generic KB developed from the PDM metadata customization project presented in section 5.5.

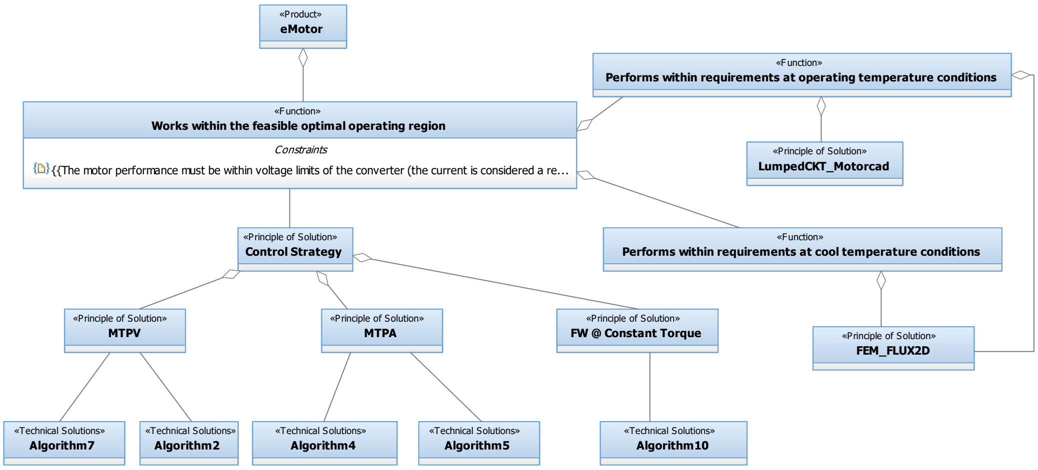


Fig. 5-10. Function diagram for eMotor PC Finder.

Note: The product model (the function diagram) by definition is used to describe knowledge about the product (parts, assembly and others). Making functions about the KBE application will transmit that the software is our product, since we want to focus on the eMotor development process our product is always the eMotor. Making the KBE application as the product is more suitable for a software development process and this can be part of the MOKA "Package" development stage not the "Formalize" stage. This was added for clarification to the reader about the product model. In particular, the pursued KBE application is more process-oriented than product-oriented. Nevertheless, a product model diagram can be used to explain the context as illustrated in Fig. 5-10.

Furthermore, the representation diagrams show the parameters within the properties of Flux (FEM tool) and Motorcad (Thermal model tool) that must be created or set for simulation and computation of performance characteristics. Fig. 3-20 shows the Flux parameters, most of these parameters are reused from the "Field Weakening FEM Coupled Algorithms" project. For Motorcad, Fig. 5-11 shows the representation of the parameters and properties used by the KBE application.

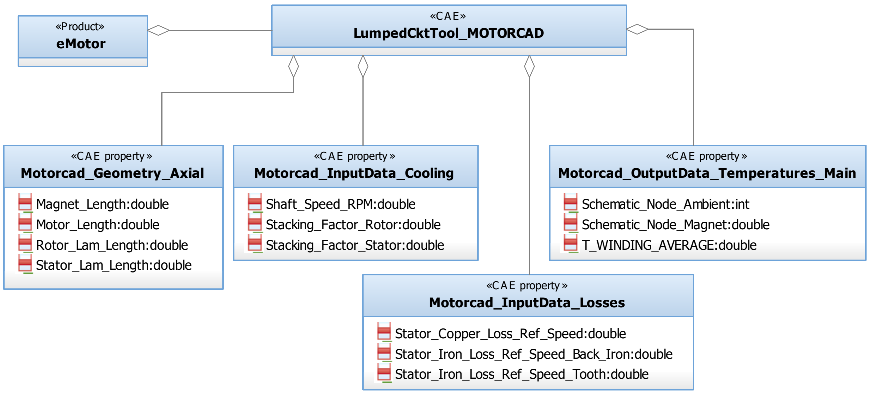


Fig. 5-11. Representation diagram for Motorcad.

The process model is now explained. Even though the process is presented in a straightforward manner, the reader must know that during the knowledge modeling many iterations were needed in order to finish it as herein presented.

Two specific-design domains are involved the electromagnetic and thermal domains. So in order to achieve the performance characteristic computations a cross-domain objective is created. The cross-domain objective is called "Compute optimal performance characteristics given torque, speed and Vlimit". The approach within this objective is the process the end-user must follow in the user interface to get the results. Fig. 3-29 shows the cross-domain objective and its approach.

The pursued KBE application will become a general design tool for brushless ac motors (IPM, SynRM, non-salient). The tool creates a dataset of results for every project. The first question that arises is; how can we make sure a simulation computation is not repeated without making the designer creating additional lines of codes during its usage? To ensure this issue do not happen, if we consider the changes that the designer can make to parameters in every electromagnetic model (making it a new model for every change of geometry, material, etc.).

Then, since every model is different, it was decided to use the Ich in addition to the input values (torque, speed and voltage) as a combination to check if a simulation has already been executed. This is why Fig. 3-29b shows indications relating to the computation of the characteristic current (Ich). As the reader already knows, the objectives allocated to the cross-domain objective must be shown in the approach. The indications in the diagram shows the allocations to these objectives. Summarizing, the designer starts by either giving the Ich in case his analysis does not involve changes to this parameter or computing the Ich following the objective "Compute Characteristic Current_IchSearch" shown in Fig. 3-26, which basically is algorithm 3. Then, the input values (torque, speed, voltage) must be entered, additionally, the motor type shall be entered since we are dealing with three types of motors.

Afterwards, the computation of performance characteristics takes place. This indication is allocated to the objective "Find requested point of eMotor for PC computation". Finally, the designer must decide if he wants to compute the performance characteristics at the operating temperature for the same inputs. This indication is allocated to the objective "Compute PC at Operating Temperature".

Following with the aforementioned objectives, Fig. 5-12 shows that from these, other objectives are derived. Fig. 5-13 presents a browser view of the objectives of Fig. 5-12 where the type and relationships can be visualized. Objectives related to field weakening characteristics computation algorithms are also used in this project (see Fig. 3-26).

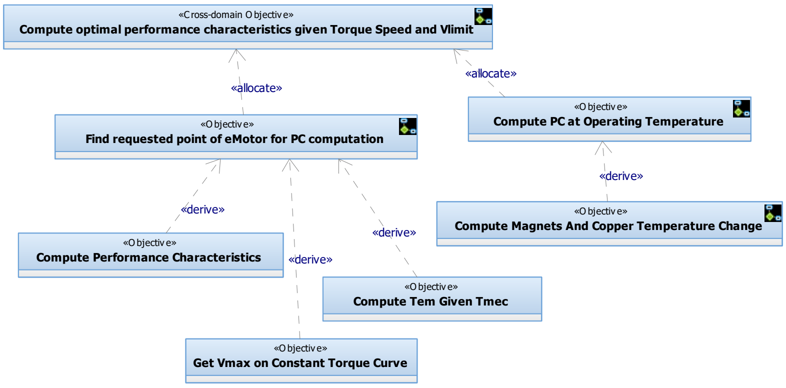


Fig. 5-12. Objectives derivation.

To avoid excessive content, only the significant diagrams will be shown to provide enough substantiation of the process.

- **FindRequestedPointOfeMotorforPCcomputation**
  - Allocations
    - ComputeOptimalPerformanceCharacteristicsGivenTc
    - Electromagnetic
  - Approach Acts
    - **ProcessthroughFEM**
  - has approachs
    - itsProcessthroughFEM
  - Indication Acts
    - ComputeMAPTForeMotorPCFinder
    - ComputeMTPVForeMotorPCFinder
  - Indication CBS
    - **UseDatasetXMLStructure**
      - Indication cnsts
        - XMLSchemaSpec
  - Satisfactions
    - Find the optimal working operation point
    - Match values of current approach

a)

- **ComputePerformanceCharacteristics**
  - Allocations
    - Electromagnetic
  - Approach CBS
    - **PCthroughAnalyticalEquations**
      - Constraint Properties
      - Indication CBS
      - Indication cnsts
        - CheckParametricDiagram
      - Parametric Diagrams
        - PCEquationsRelationships
  - Block Definition Diagrams
    - PCParameters
  - Derivations
    - FindRequestedPointOfeMotorforPCcomputation
  - has approachs
  - Satisfactions
    - Compute performance characteristics
    - Match values of current approach

b)

- **ComputeTemGivenTmec**
  - Allocations
  - Approach CBS
    - **Tmec2TemEquations**
      - Binding Connectors
      - ConstraintParameters
      - Indication cnsts
        - ElectromagneticPower
        - ElectromagneticTorque
  - Derivations
    - FindRequestedPointOfeMotorforPCcomputation

c)

- **GetVoltageOnConstantTorqueCurve**
  - Allocations
    - Electromagnetic
  - Approach CBS
    - **FEMCoupledAlgorithm_VoltOnTorque**
      - Indication cnsts
      - is approach ofs
  - Block Definition Diagrams
  - Criterias
    - Algorithm10Description
  - Derivations
    - FindRequestedPointOfeMotorforPCcomputation
  - has approachs
    - itsFEMCoupledAlgorithm_VoltOnTorque
  - Satisfactions

d)

- **ComputePCatOperatingTemperature**
  - Allocations
    - ComputeOptimalPerformanceCharacteristicsGivenTorqueAndSpeedAnd\
    - Electromagnetic
  - Approach Acts
  - Satisfactions
    - Compute performance characteristics at operating temperature

e)

- **ComputeMagnetsAndCopperTemperatureChange**
  - Allocations
    - Thermal
  - Approach Acts
  - Derivations
    - ComputePCatOperatingTemperature
  - Indication CBS
    - **CopperAndMagnetsTemperatureChange**
      - Binding Connectors
      - ConstraintParameters
      - Indication cnsts
        - dTcu
        - dTm

f)

Fig. 5-13. Browser view of objectives.

Beginning with the derived objective “Compute Tem Given Tmec”, this is an objective with an approach CB type holding the equations to pass from the mechanical torque given by the designer to the electromagnetic torque that the electromagnetic model handles. Once this is done, we have the inputs to find the requested point in one of the three optimal regions. Fig. 5-14 reveals the main process of the application in FEM (approached contained by the objective of Fig. 5-13a).

The core of the process is, when a point is requested the algorithm starts searching the point in the MTPV region. If the found torque is lower than the requested torque then the requested torque cannot be reached with the voltage limit and the user only gets the performance characteristics (PC) of this maximum torque operating point for the given voltage limit with a warning that the requested point cannot be reached. If the torque is equal to the requested torque then the user gets the PC of the requested point in the MTPV region. On the other hand, when the torque is greater than the requested torque then it is searched in the MTPA region. When found, if the voltage is lower or equal to the voltage limit then the PC are presented. Otherwise, the optimal point is searched in the FW region along the constant torque curve where the voltage limit is met and the PC data is presented.

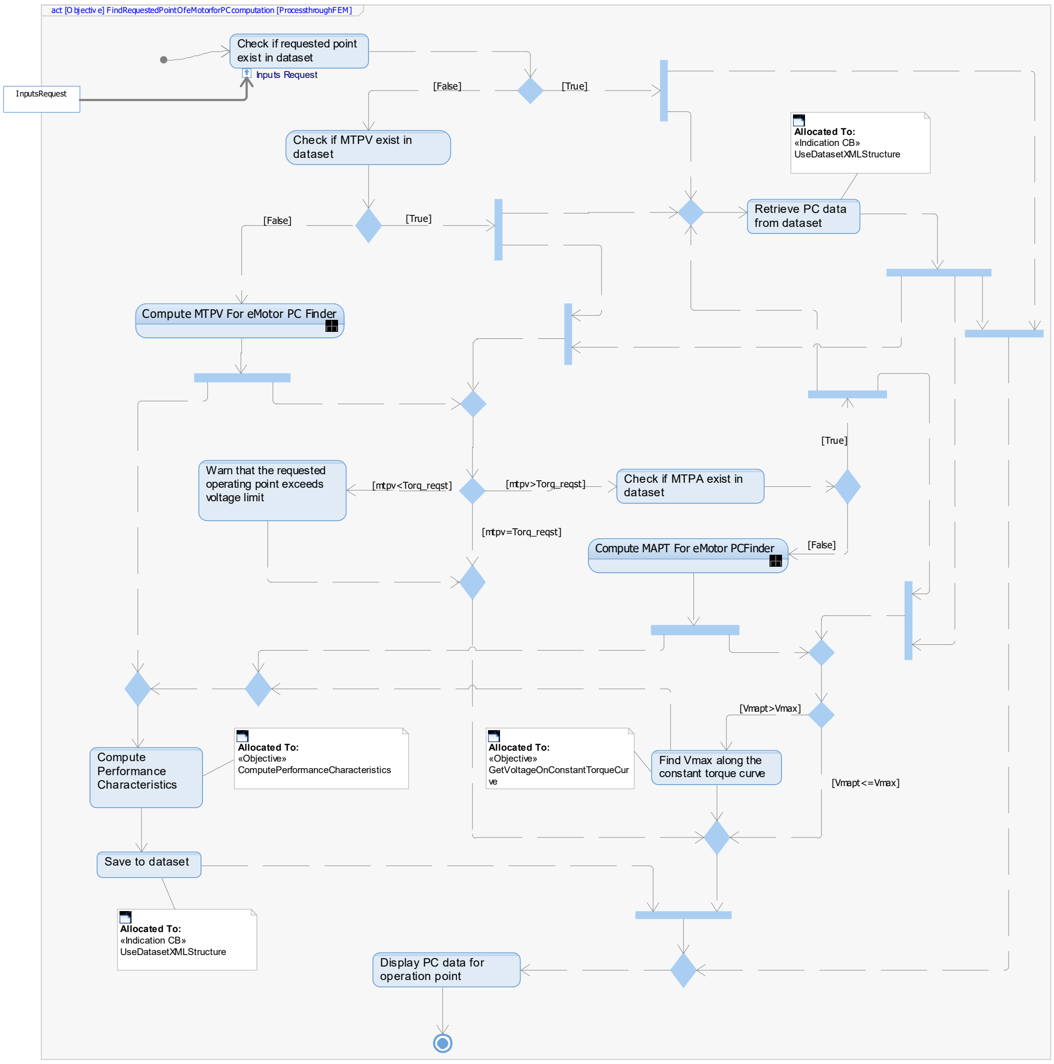


Fig. 5-14. Main process of eMotor PC Finder.

The rest of the indications in Fig. 5-14 are indications regarding data management and not repetition of the same simulation point, considering also not repeating simulations during the search process. In the approach Act diagram two indications call an expanded indication Act, these are “Compute MTPV for eMotor PC Finder” and “Compute MAPT for eMotor PC Finder”. Both are shown in Fig. 5-15. The indications filter the way to compute the requested point by applying a different algorithm in the process for faster convergence depending whether or not the motor is non-salient.

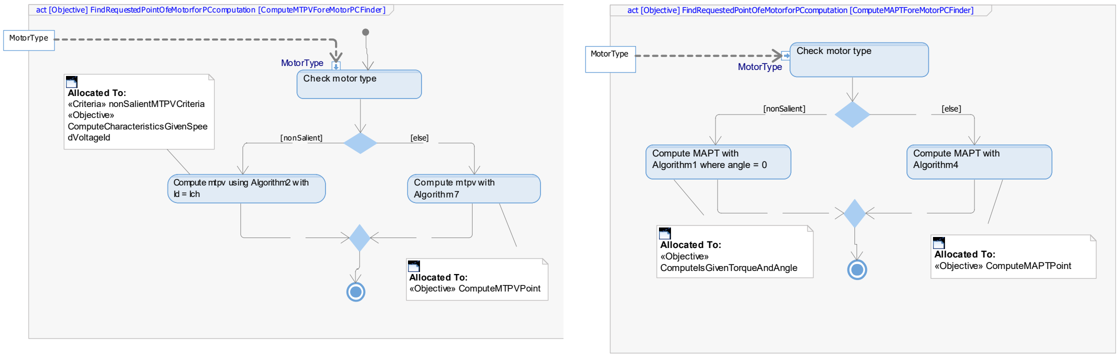
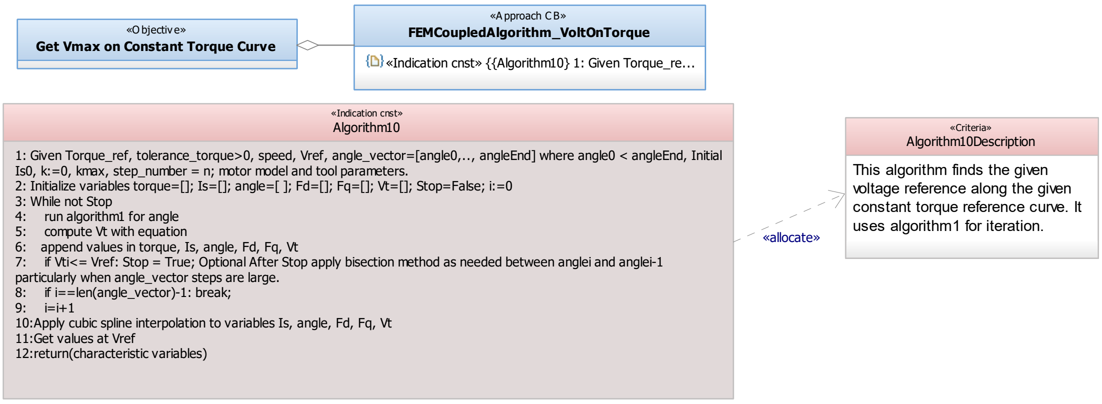


Fig. 5-15. Expanded indications.

An additional algorithm had to be developed for the indication "Find Vmax along the constant torque curve". The indication is allocated to the objective containing the algorithm and it is presented and described in Fig. 5-16.



```
1: Given Torque_ref, tolerance_torque>0, speed, Vref, angle_vector=[angle0,.., angleEnd] where angle0 < angleEnd, Initial Is0, k:=0, kmax, step_number = n; motor model and tool parameters.
2: Initialize variables torque=[]; Is=[]; angle=[ ]; Fd=[]; Fq=[]; Vt=[]; Stop=False; i:=0
3: While not Stop
4:     run algorithm1 for angle
5:     compute Vt with equation
6:    append values in torque, Is, angle, Fd, Fq, Vt
7:     if Vti<= Vref: Stop = True; Optional After Stop apply bisection method as needed between anglei and anglei-1 particularly when angle_vector steps are large.
8:     if i==len(angle_vector)-1: break;
9:     i=i+1
10:Apply cubic spline interpolation to variables Is, angle, Fd, Fq, Vt
11:Get values at Vref
12:return(characteristic variables)
```

This algorithm finds the given voltage reference along the given constant torque reference curve. It uses algorithm1 for iteration.

Fig. 5-16. Algorithm 10: Find Vmax along constant torque curve.

Regarding the objective "Compute performance characteristic", it contains an approach CB type holding all the equations to get the performance characteristics of the motor. The indications CB with the equations are shown in Fig. 5-17.

The objective "Compute PC at operating temperature" related to the thermal domain is shown in Fig. 3-28. According to the process in Fig. 3-29, in order to use this thermal related objective previously the computation in cool conditions had to be executed. The indications of the approach (see Fig. 3-28) are first compute the copper and magnets temperature change, this indication is allocated to the derived objective "Compute magnets and copper temperature change" presented in Fig. 3-27.

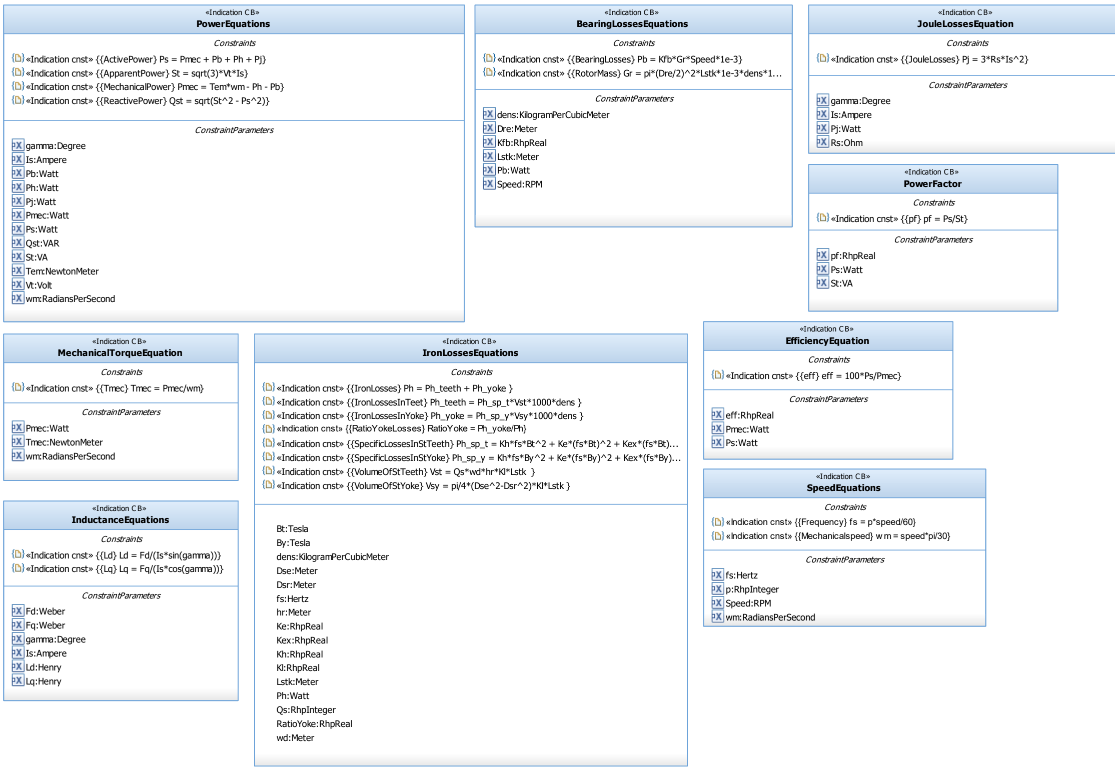


Fig. 5-17. Indications CB with equations.

The main purpose of the objective “Compute magnets and copper temperature change” is to simulate in Motorcad with the resulting iron and Joule losses in cool conditions the change in temperature of magnets and copper. Once this is finished, the second indication tells us that we retrieve this temperature changes and compute the new resistance (Rs) and the residual magnetic flux density (Br) value. This indication is expanded with an indication CB type containing the equations to do so, see Fig. 5-13f. Afterwards, the new Rs and Br values are set in the electromagnetic model and the process of finding the optimal point is repeated because the point has been displaced in the electric current phasor circle diagram.

All the data is saved in XML format following the indication CB “Use dataset XML structure” containing an indication with the schema presented in Fig. 5-18. Finally, to illustrate refinement with SysML some diagrams are created to support the Package stage by linking knowledge and software modules. For instance, the modules of the KBE application are represented by SysML blocks see Fig. 5-19. The modules with their respective objectives as parts are shown in Fig. 5-20. Furthermore, to show the relationships between parameters from the software modules, the electromagnetic and thermal design tools and the PC equations shown in Fig. 5-17 a parametric diagram is created providing clarification on how these equations should be executed. The parametric diagram is shown in Fig. 5-21.

```
{<?xml version="1.0" encoding="utf-8"?>
    <Project>
         <ProjectName> </ProjectName>
                <MotorcadName> </MotorcadName> 'if motorcad model file path provided
                <DateOfCreation>YYYY-MM-DD TIME</DateOfCreation>
       <InputsReq id="[Torque, Speed, Voltage, Ich]" id2="[Torque, Speed, Voltage, Ich]">
         <Tmec> </Tmec>
         <Is> </Is>
         <Gamma> </Gamma>
         <Vt> </Vt>
         <Speed> </Speed>
         <Ich> </Ich>
         <Pmec> </Pmec>
         <Ps> </Ps>
         <Pj> </Pj>
         <Ph> </Ph>
         <Pb> </Pb>
         <pf> </pf>
         <eff> </eff>
         <Fd> </Fd>
         <Fq> </Fq>
         <Ld> </Ld>
         <Lq> </Lq>
         <Rs> </Rs>
         <Lfc> </Lfc>
         <Qst> </Qst>
         <Tem> </Tem>
         <Ksl> </Ksl>
         <Br> </Br>
         <dTcu> </dTcu>
         <dTm> </dTm>
         <mode> </mode>
         <ratio_yoke> </ratio_yoke>
       </InputsReq>
    </Project>}
```

Fig. 5-18. XML schema to save PC.

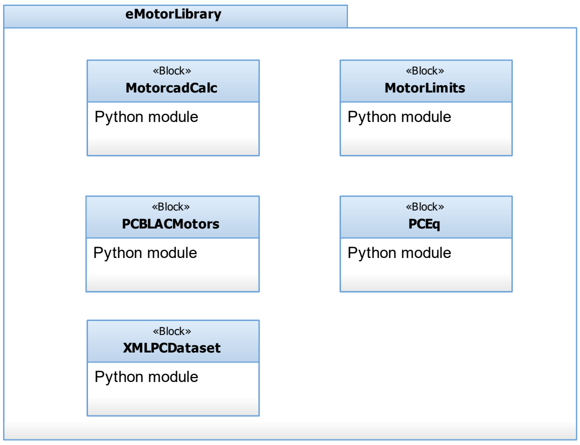


Fig. 5-19. SysML refinement: Python package containing the modules to execute the KBE application.

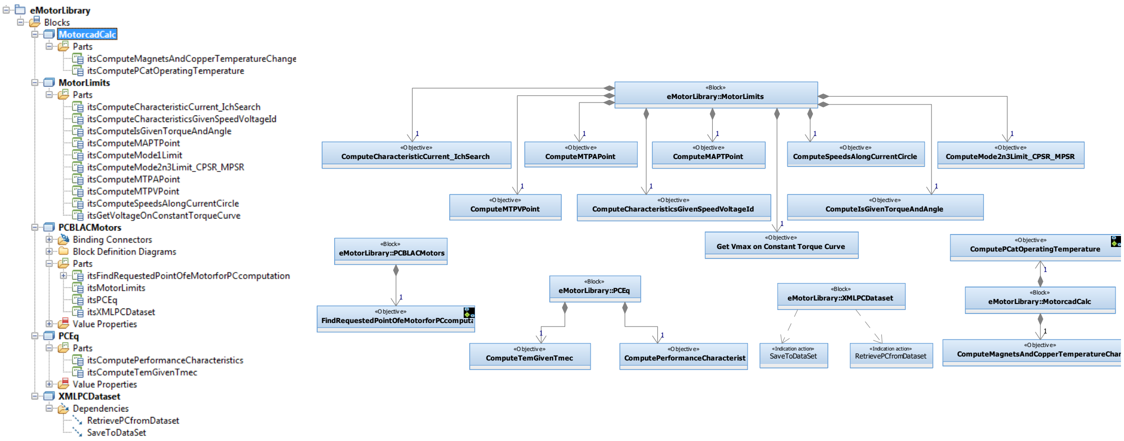


Fig. 5-20. SysML refinement: Objectives each module must accomplish.

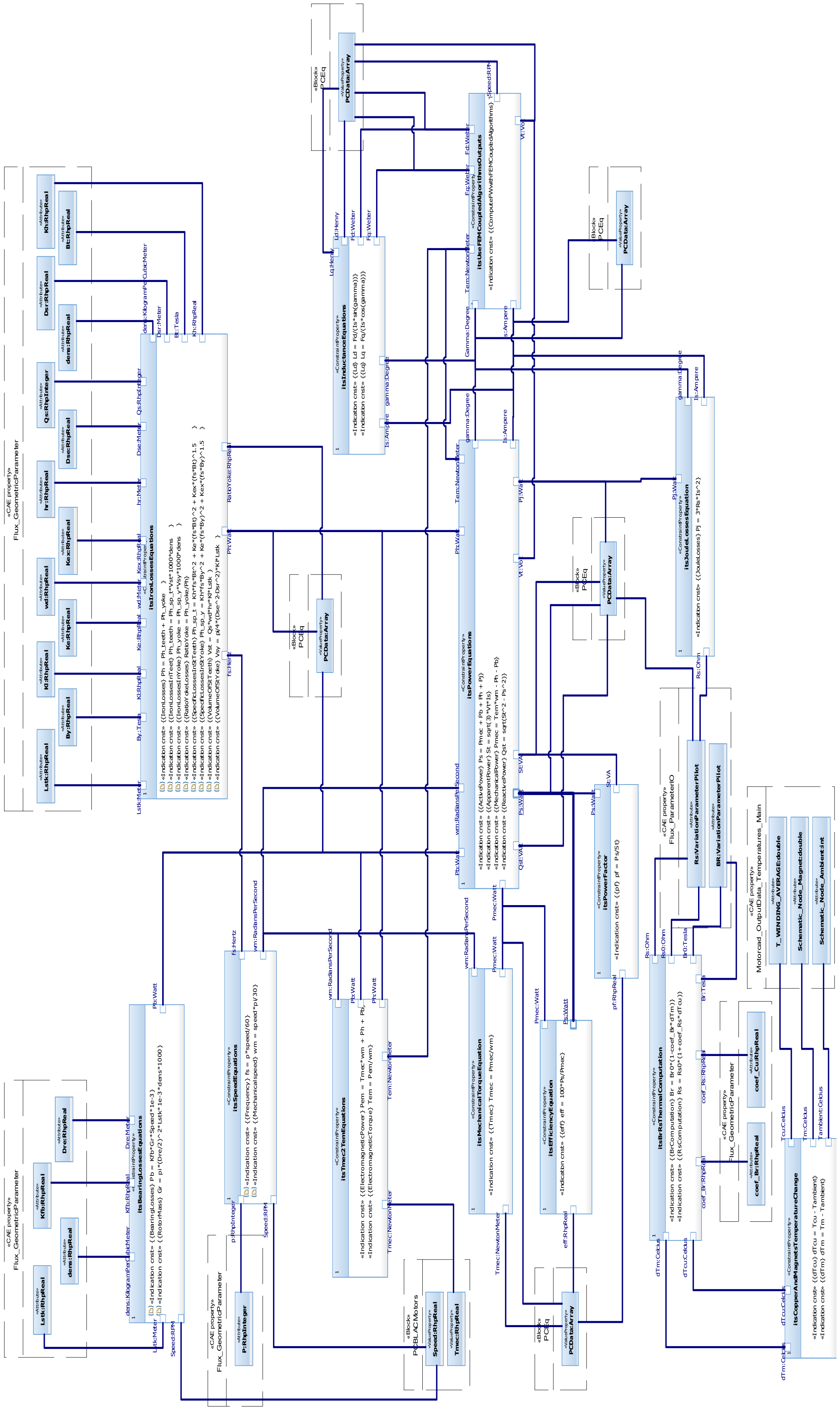


Fig. 5-21. SysML refinement: Parametric diagram showing tools properties with equations.

5.3.3 Package

In this stage, the KBE application is developed, the formalized knowledge is transformed into code. The modules containing the formalized knowledge were shown in Fig. 5-19. The programming language used to achieve the application are Jython for the FEM tool and VBscript for the thermal lumped circuit tool. It was decided to implement the KBE application in Flux 2D (the FEM tool) since it offers an environment to run Jython scripts or modules for automation of tasks. That advantage is taken to run the KBE application within the same FEM tool and create a middleware to run Motorcad from Flux when requested.

The modules are developed and classified as seen in Fig. 5-22. The knowledge contained in each module is as shown in Fig. 5-20. XMLPCDataset is used for storage and retrieval of data from the XML files following no specific represented knowledge that is why is considered and auxiliary module. CallMotorcadCalc and main_PCFEM are modules that serve for connection between both tools. They call specific methods from the knowledge wrappers modules.

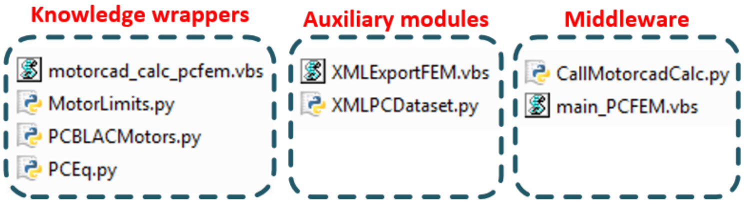
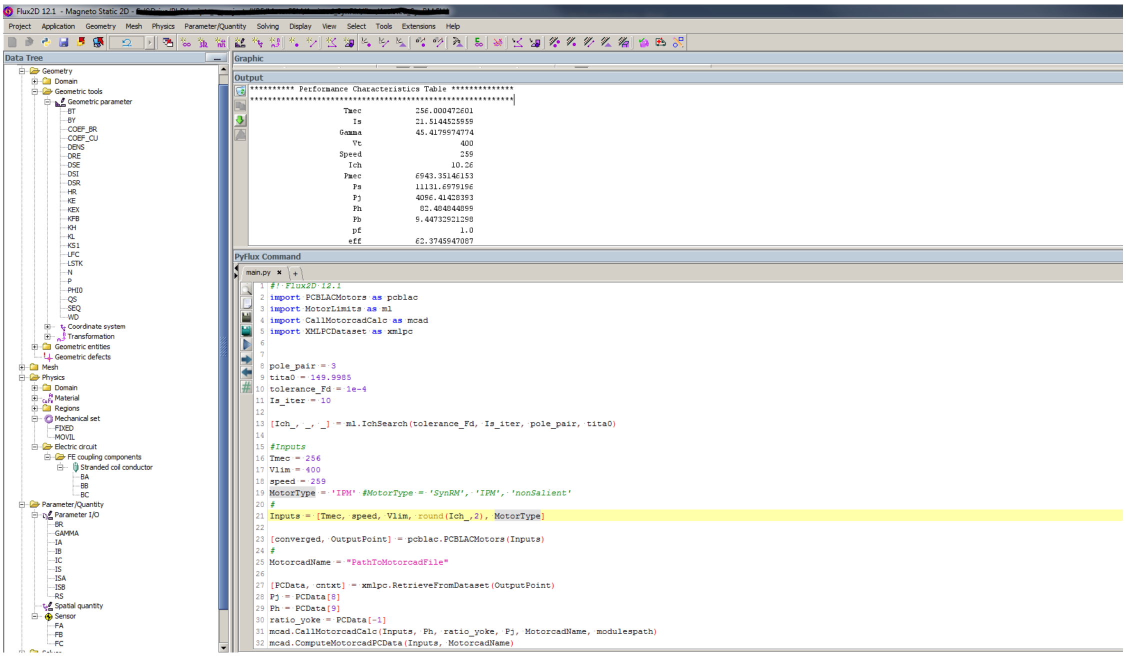


Fig. 5-22. Classification of modules.



```
********** Performance Characteristics Table **************
**********************************************************
                   Tmec          256.000472601
                     Is          21.5144525959
                  Gamma          45.4179974774
                     Vt                    400
                  Speed                    259
                    Ich                  10.26
                   Pmec          6943.35146153
                     Ps          11131.6979196
                     Pj          4096.41428393
                     Ph           82.484844899
                     Pb          9.44732921298
                     pf                    1.0
                    eff          62.3745947087
```

```python
#! Flux2D 12.1
import PCBLACMotors as pcblac
import MotorLimits as ml
import CallMotorcadCalc as mcad
import XMLPCDataset as xmlpc


pole_pair = 3
tita0 = 149.9985
tolerance_Fd = 1e-4
Is_iter = 10

[Ich_, _, _] = ml.IchSearch(tolerance_Fd, Is_iter, pole_pair, tita0)

#Inputs
Tmec = 256
Vlim = 400
speed = 259
MotorType = 'IPM' #MotorType = 'SynRM', 'IPM', 'nonSalient'
#
Inputs = [Tmec, speed, Vlim, round(Ich_,2), MotorType]

[converged, OutputPoint] = pcblac.PCBLACMotors(Inputs)
#
MotorcadName = "PathToMotorcadFile"

[PCData, cntxt] = xmlpc.RetrieveFromDataset(OutputPoint)
Pj = PCData[8]
Ph = PCData[9]
ratio_yoke = PCData[-1]
mcad.CallMotorcadCalc(Inputs, Ph, ratio_yoke, Pj, MotorcadName, modulespath)
mcad.ComputeMotorcadPCData(Inputs, MotorcadName)
```

Fig. 5-23. Screenshot of Flux properties and KBE application user interface

The design tools must be set as the representation diagrams Fig. 3-20 and Fig. 5-11, for instance, Fig. 5-23 shows a screenshot of Flux 2D and its properties. The same figure presents the user interface, a main script is loaded in the PyFlux Command window where the user can import the modules and follow the process of Fig. 3-29. The example shows the modules that need to be loaded as well as the commands for execution. For this case, the Ich is computed. The output windows shows an example of a table displayed to the user but as mentioned in the *formalize stage* the data is saved in an XML file too, following the structure of Fig. 5-18. In addition, a different XML file is created for thermal computation. This was decided to facilitate the data classification for the designer.

Moreover, due to the core of the process computation is mostly done through the algorithms described in the previous micro-level project. Then, the same validation in Appendix B applies for this KBE application.

5.3.4 KBE application use case example: Performance comparison between a SynRM and PMSynRM

The developed KBE application was developed with general knowledge about brushless motors; therefore, it can be applied in different design scenarios. However, here is a simple but illustrative example of its usage by making a comparison study of performance between a ferrite-assisted SynRM (technically IPM) and a SynRM motor.

For this case, the same SynRM electromagnetic and thermal models are used with the difference the PMSynRM has ferrite magnets inserted in its slots. Table 5-1 shows the parameters for both motor models. Three points of operation are analyzed in order to compare their performance. Table 5-4 presents these operation points, through the KBE application both models were executed in parallel with the instructions to run the operation points following the same order presented in the table.

Table 5-4. Points of operation for comparison study and computational time of execution.

	Torque	Speed	Voltage limit	Temperature	SynRM exec. time	PMSynRM exec. time
Max.	416 Nm	166 RPM	400 V	ambient	23 min	46 min
Rated	256 Nm	166 RPM	400 V	ambient	24 min	20 min
RMS	175 Nm	166 RPM	400 V	ambient	21.92 min	22.6 min
RMS	175 Nm	166 RPM	400 V	operating	50 min	50 min
	-	-	-	**Total:**	118.92 min	138.6 min

For the maximum point, the results showed that the SynRM was not able to reach that operating point, then, only the MTPV point was returned as result. On the contrary, the PMSynRM can give it but on the FW region because in the MTPA curve the voltage limit was surpassed.

A summary of results for each requested operating point is given in Table 5-5. From the results, it can be concluded that this micro-level KBE application can be used for rapid comparison of different designs at the same working point obtaining automatically the electrical and thermal performance.

The computational time of execution were provided in Table 5-4. The time of execution depends on many factors such as the electromagnetic model, type of motor, and if distributed computing is used. The algorithms were processed with an Intel core i5-4590 CPU with 16 GB of RAM and no distributing computing was used. Thus, there is still a margin of making the computational process faster. The algorithms perform faster for non-salient motors because their behavior is predictable. Despite the aforementioned facts, FEM always consumes many computational resources. However, one advantage of the KBE application is that the designer only had to insert the commands of execution to check later the results, thus with no effort for computation of parameters allowing using that time for other tasks.

Table 5-5. Performance of SynRM and PMSynRM.

Point of Op.	**SynRM**					**PMSynRM**				
Nm@RPM@V	**Tmec**	**Is∠gamma**	**Vt**	**Eff**	**Mode**	**Tmec**	**Is∠gamma**	**Vt**	**Eff**	**Mode**
416@166@400	386	20.49∠77.4	400	63.94	MTPV	416	15.25∠66.1	400	77.29	FW
256@166@400	256	12.01∠61.4	379.1	76.78	MTPA	256	9.75∠53.8	364.74	83.02	MTPA
175@166@400	175	8.87∠56.5	346.7	79.91	MTPA	175	7.18∠48.6	331.65	85.30	MTPA
175@166@400 (@op. temp.)	175	8.87∠56.5	364.33	72.83	MTPA	175	7.30∠49.5	344.55	80.72	MTPA
dTcu	135.19 °C					100.82 °C				
dTm	-					84.00 °C				

5.4 Discussion

The implementations presented in this chapter provides substantiation of the proposed KBE system framework following the KBV2-model for the micro level cases. After the implementation, the reader could noticed that in order to develop KBE applications the knowledge base generation process is crucial for their successful development. In addition, the knowledge is kept in models for future inspection or knowledge reuse.

The implementation of the framework provides advantages for repetitive task reduction. It was found that the computational time reduction for one of the worst-case scenario of its usage regarding the KBE application *eMachine operating limits* is 30% less than the normal process followed by the designer without the KBE application. Being able to optimize its use even more. Furthermore, it should be considered that the tasks required less human work. Therefore, the KBE application successfully accomplished the goal of supporting the designer to compute complex points faster and with less work effort.

The developed KBE application *eMotor Performance Characteristics Finder* integrated the aforementioned application, extending its capabilities. Therefore giving example to some extent of the MVP principle during development. A complete detailed of the KBE application was provided through the formalize stage. Models for each type of the knowledge representation following the proposed micro-process metamodel were provided. Likewise, examples of refinement through SysML diagrams were also given.

In conclusion, the KBES framework micro-level proposal is considered successful since following the process and using the new micro-level knowledge representation allowed the creation of a KBE application with the capability of computing complex points of operation with little effort from the designer and provides the results in acceptable times of execution.

Chapter 6

MACRO-LEVEL IMPLEMENTATION

The purpose of this chapter is to provide substantiation of the proposed KBES framework for the macro-level implementation. A KBE system is developed for the Industrial V-cycle. The domain of application is the elevator industry. This chapter shows a practical implementation of the KBES framework to develop a KBE system for the design of a viable motor solution.

The chapter is composed of the following sections. Section 6.1 introduce the chapter Section 6.2 describes the development of the KBE system. Section 6.3 implements the KBE system for the design of viable eMotor showing the process followed for each of the stages of the Industrial V-cycle. Finally, section 6.4 discusses the results.

6.1 Introducción

This chapter provides a practical macro-level implementation example of the KBE system framework. The implementation is carried out for the Industrial V-cycle regarding the design of electric motors for elevator systems. A KBE system is developed by integrating technologies to enhance the development process. Attention is centered in the macro-level KBE application. The implementation gives a complete description on how the KBE system works and how it is developed with description of tools, architecture, user interfaces, and enclosed micro-level applications (explaining the knowledge behind them). The section finishes by showing systematically the implementation of the KBE system.

The process herein undertaken is based on previously validated motors. The only degrees of freedom for the design of a new motor are the length of stack and the coil construction model. In other words, the designer must satisfy the requirements for an elevator system by adjusting the length of stack of the motor and the number of turns of coils with the flexibility of adjusting the number of wires in parallel, wire gauge and type, and location of wires in the slot. The design is validated for the electromagnetic and thermal domains through their respective models keeping the same geometry, materials, layouts and manufacturing processes.

In addition, a comparison of estimated time consumption with and without the KBE system is given. This chapter will not give details about how to design an electric motor.

6.2 A KBE system for the development of eMotors for the elevator industry domain

This section presents the development of a KBE system for the Industrial V-cycle. The KBE system is created for the elevator industry domain and its main goal is to provide a viable eMotor solution based on previous validated designs and manufacturing processes that are used as references to satisfy new requirements for an elevator system. This section explains how the KBE system works with emphasis on the macro-level KBE application including short descriptions of the integrated micro-level applications for each stage in the Industrial V-cycle.

The KBV2-model for the KBE system is presented in Fig. 6-1. The application integration testing stage is highlighted in yellow to remark and remind the KBE application has no direct interaction with this stage. Another remark is that only the electromagnetic and thermal domains are directly involved in the design process. This does not mean that the other domains are not considered, since the system works with previous validated motor models the rest of domains and manufacturing processes are implicitly included when the selection of functions and modules take place. Assuring the right performance of the motor for each design domain, further explanation during the description of the system.

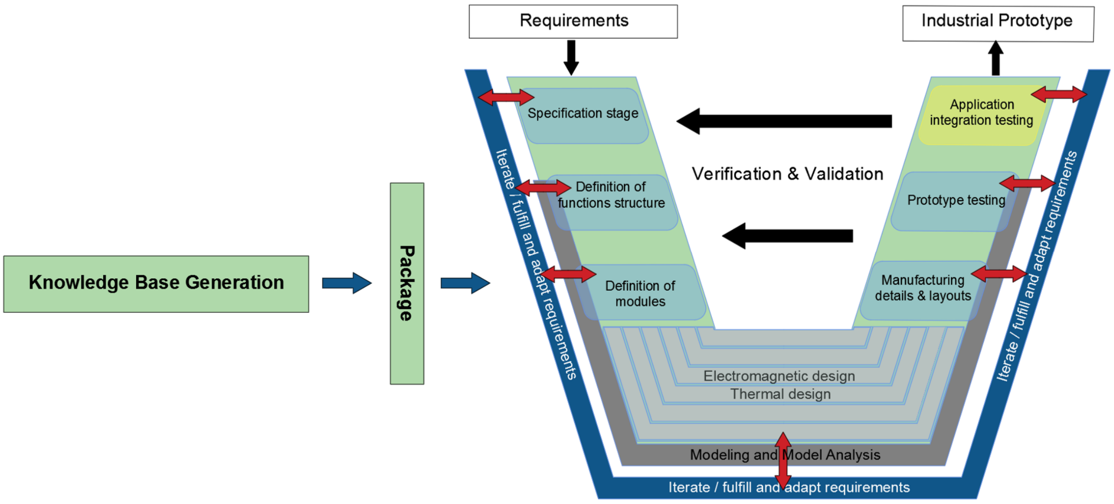


Fig. 6-1. KBV2-model of the KBE system for the development of eMotors for the elevator industry.

6.2.1 Problem context

The case study is carried out in an electric machine design department. It has been identified the need to provide fast viable solutions of electric motors that adapts to specific elevator system requirements by customizing validated models of motors with define manufacturing processes. It was concluded that designers follow repetitive and well stablished technical steps to accomplish the goal for a viable motor solution given a set of requirements that depends on a particular elevator system. In the current process there is no data management system, thus, the information is scattered and its retrieval is a concern since the data is hard to find or sometimes even its location forgotten, therefore, slowing the process. In the previous chapter, it was mentioned that studies say that an engineer to retrieve information may take up to 70% of its time effort. Furthermore, each designer retains its own knowledge to provide a solution making the transfer of knowledge to new co-workers complicated and time consuming. For all these reasons, the creation of a KBE system solution suits well the problem context.

The outcome of the KBE system is a feasible industrial eMotor prototype, ready for serial or mass production. Recalling the industrial prototype V-cycle from Chapter 2, the electric motor developer is limited in variation of parameters for a finished motor. In other words, most components are reused especially those that favors manufacturing such as the stamping die, winding process and materials.

The KBE system shall be able to store, track and trace the entire development process by integrating in an organized and collaborative way all the technologies involved in the development process. The KBE application shall have the capability of automatically generating motor alternatives varying only the number of turns and length of stack of the motor. In addition, it shall provide a guideline to support the designer in each stage and have at hand the knowledge expertise in case the designer wants to understand in more detail the process. The KBE application shall have the capability of tracking the time spent by the designer in each stage as well as registering the events the end-user makes. It shall have the capability of ensuring a viable solution for the specified elevator system.

6.2.2 Formalize

This subsection presents the macro-level knowledge of the KBE system. This knowledge is also available through the UI to the end-user in the KBE application. The knowledge models for the macro-level view is the system designed to create a motor solution in an organized manner with the integration of the technologies reviewed to enhance the product development process. The KBE application as part of the KBE system, integrates six micro-level applications. The knowledge model of only one of these micro-level KBE applications considered the core of the process is described in this subsection. The rest of KBE applications will be briefly described.

6.2.2.1 KBE system description

Fig. 6-2 presents the selection of tools as part of the devised KBE system. For the PDM Aras innovator v11.0 and Synect v2.4 were chosen. Aras to manage bill of materials i.e. CAD related models in addition to manufacturing processes related to particular parts, assemblies or products. Synect is used for data management of domain-specific design models as well as the integration of models. Synect as a tool is not considered entirely a PDM system but can be seen as an extension of any PDM system to support data management of models. Its capabilities are taken in advantage to extend Aras Innovator. For this particular case, only the electromagnetic and thermal models are implemented. The PDM system is considered the source of truth of the KBE system, thus all information updated in PDM must be used throughout the project.

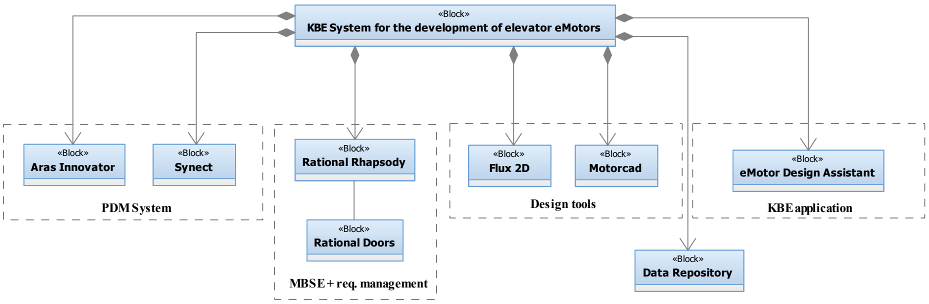


Fig. 6-2. Tools selection for the KBE system.

Rational Rhapsody v8.3.1 is the tool that holds the system models to facilitate the end-user the tracing and monitoring of the process for the Industrial V-cycle including requirements verification. Please recall that Rational Rhapsody is also the tool to formalize the knowledge. Linked to Rhapsody is Rational Doors v9.5.2.2, this is a well-known requirement management tool. The KBE system is developed for low-level requirements, therefore, Rhapsody is the tool to trace these requirements and update Rational Doors. In this work, these two tools are the only ones integrated outside the KBE application.

The design tools to perform simulations of models are Flux 2D v.12.1 for the electromagnetic models and Motorcad v.11.1.23 for thermal lumped circuit models. The data repository was left intentionally as generic because several software tools are used such as SQL databases and cloud storage solutions, there is flexibility on the way to handle the data objects. However, the main purpose of the data repository is to provide a common repository for the KBE system in order to provide a solution of common data shared among the stakeholders. Finally, the KBE application called eMotor design assistant is the main part of the KBE system. The KBE application is in charge of supporting the eMotor developer through the entire system processes.

Besides Raphsody and Doors some of these tools can be integrated among themselves, for instance, Doors can also be linked to Synect for requirement management supporting test cases management. This can be used to leverage the company resources. Nonetheless, the integration among tools is out of the scope of this work, only the integration of the tools with the KBE application will be covered.

Fig. 6-3 summarizes the main actions of the KBE system. The system is composed of many diagrams. However, this diagram is presented because it shows and summarizes the system high-level actions with the purpose of introducing it to the designer. Although it does not show all of the features of the KBE system for instance the loop among the V-cycle actions (specification stage, definition of functions structure, definition of modules, etc.), micro-level activities and others. It will be used to explain the main aspects of the system. Inside each action of the V-cycle more activities are carried out, that is the reason each action is of the call behavior type. The rest of the features of the KBE system will be described in Section 6.3 for better contextualization. The KBE system is described emphasizing the KBE application as the main tool of the system.

Following the diagram of Fig. 6-3, starting in the specification stage the KBE application needs the elevator system parameters in order to compute the low-level requirements values. The low-level requirements were derived following the requirement engineering framework presented in Chapter 2. The elevator parameters are saved in Synect and exported with Matlab extension format for the KBE application. Once the low-level requirements values are generated, these are set in Rhapsody and because of the linked by Gateway to Doors, they are automatically updated in Doors too. This facilitate requirements traceability on the whole development process. Furthermore, an XML file containing the Synect structure is generated in order to load these requirements in Synect for test cases management during the design and prototype testing phases. The next step is the definition of functions structures, since we are working with validated motor features this stage provides the properties of the motors from the product portfolio previously set as references for elevator systems. By doing this, the end-user is limited to select features of motors that can lead to a possible solution. In other words, we are limiting the design space of solutions.

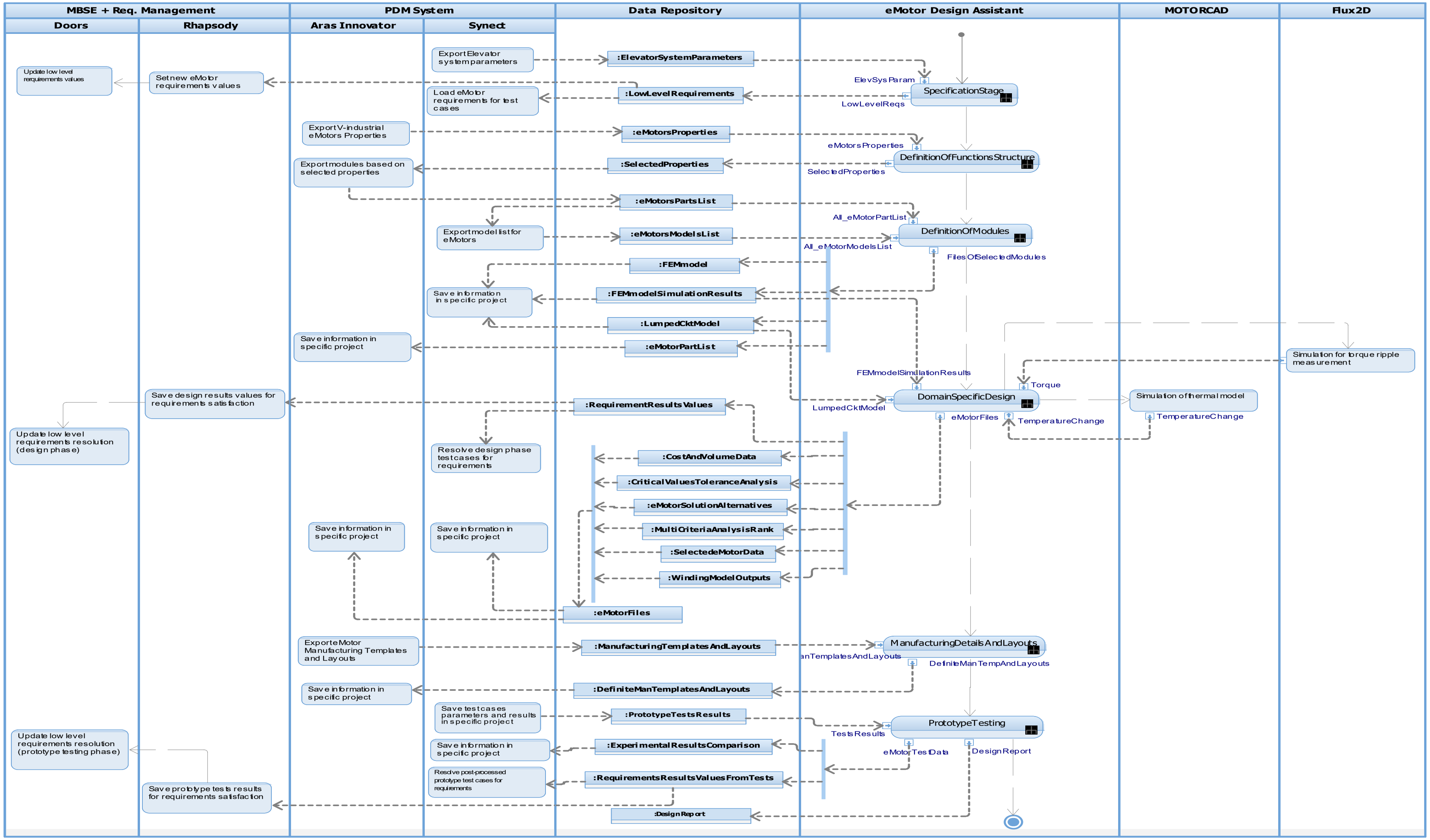


Fig. 6-3. KBE system activity diagram.

The properties available in the eMotor product portfolio are exported from Aras innovator and imported by the KBE application for the end-user to select among them. The selected properties in the KBE application are saved in the data repository for future references for the end-user.

The next stage is the definition of modules. With the selected properties from the previous stage the part list of each possible motor solution is exported from Aras innovator. The motor stator and rotor lamination are extracted from these part lists. Similarly, from Synect the models lists linked to the part list of each motor is exported. The designer (end-user) defines the modules to use and the KBE application provides the user the files of the selected motor.

The files in the repository are:

1. FEM model: in Flux format.
2. FEM model simulation results: this file contains previously made simulations for a complete characterization of the motor in the FEM tool. Check ModeladoScript data format in Appendix D for more information. A tool to export these data from Flux was developed, the output file format is XML although initially an Excel file was used, so both files format are supported by the KBE application but because of the developed tool, Excel files will be no longer required in the future.
3. Lumped circuit model: this file is the model of the motor in Motorcad.
4. eMotor part list: is the BOM of the selected motor in Microsoft excel format.

A specific project is created in Aras and Synect and the information is saved in each system. The next stage is the domain-specific design phase, speaking in general terms, this stage uses the FEM model simulation results and the lumped circuit model to create motor alternatives based on the designer inputs. The design process also involves simulations in Flux and Motorcad, the former with the purpose of checking the torque ripple requirement satisfaction and the latter to obtain temperature changes of copper and magnets in operating conditions. At the end of the process, the designer selects a motor solution and the KBE application creates a set of files describing the selected motor solution and saves it to the working path of the project in the repository. The files includes:

1. Cost and volume data of all possible solutions.
2. All solutions alternatives from which one should be selected.
3. Multi-criteria decision making (MCDM) analysis (if used).
4. Performance data of the selected motor. The selected eMotor data includes the performance, no load, load and cost and volume tables for the selected solution.

5. Tolerance analysis, a file is created with the critical values of the analysis. The information is saved in the specific project in the PDM system and the test cases for the design phase are resolved.

Once the design phase finishes, the manufacturing details and layouts stage comes into place. The manufacturing templates and layouts are pre-established forms and drawings that comply the manufacturer requirements in order to proceed with the manufacturing of the prototype. The command of downloading the files is given in the KBE application but in the background they are imported from Aras and once the information is filled the definite files are output to the working path of the repository.

Finally, the last stage involving the KBE application is the prototype testing stage where the results are imported following the same template created in the design phase and the KBE application makes a comparison between FEM and experimental results. The designer must filled the results values to check if they satisfy the requirements. These requirements are added in Rhapsody and updated in Doors. The KBE application finishes with the generation of a design report for the eMotor developer.

There is still one more phase missing, the application integration testing stage. The KBE application offers no functions for the eMotor developer during this stage. However, the stage must be carried out in order to finish the validation of the expected industrial prototype.

The KBE system has been described. The macro-level KBE application integrates six micro-level KBE applications. These applications are now briefly described; later more information will be given during the KBE system implementation section.

1. ElevSys: Based on the requirement engineering knowledge regarding elevator systems, this application generates the values of the low-level requirements given the elevator input parameters.
2. FindValidN: This application finds the solution alternatives sweeping the number of turns for each length of stack given in array form. It uses the withheld knowledge to analyze both, ambient and operating conditions searching and making decisions on whether or not the given set of parameters in every iteration can be considered a motor solution alternative. It uses ModeladoScript to compute the performance of each motor in ambient and operating conditions.
3. ModeladoScript: The application computes the performance values of a given motor, the only degrees of freedom are the length of stack and number of turns for each motor. The motor output performance is obtained through the given input parameters (rated torque, peak torque, RMS torque, speed and voltage limit), FEM simulation results and

Motorcad file. It uses Motorcadcalc to get the change of temperature for operating conditions computation. Motorcadcalc is a function that automatically computes an array of temperature changes using the lumped circuit model given an array of Joule and iron losses.

4. ModeladoScriptTolerance: This application analyzes statistical distributions of the machine performance in function of three design variables selected from a sensitivity analysis study. These design variables are the magnet remanent field (Br), stacking factor Kl and overlapping factor (Kov). The application considers serial or mass-production tolerance variations, thus obtaining critical values that supports the designer on the decision whether the design is valid or not. This application uses as basis ModeladoScript, this KBE application is an example on how the end-user can expand the features of an application mentioned in the previous chapter. The creator of this application was provided with ModeladoScript, he used it and later adapted it to create ModeladoScriptTolerance [91].
5. WindingComp: It uses the information stored in each motor reference to compute winding parameters based on the selected stator and rotor geometry following the experts' knowledge.
6. MCDMAnalysis: This application uses multi-criteria decision making (MCDM) methods to support the designer on the best motor selection from all alternatives. One of four methods can be used these are, technique for order preference by similarity to ideal solution (TOPSIS), weighted sum model (WSM), weighted product model (WPM) and weighted aggregated sum product assessment (WASPAS). The methods were customized for accurate performance score computation for each alternative from a set of chosen motor performance criteria allowing the user to select the best motor from the output rank.

Moreover, the micro-level knowledge on how to execute each stage successfully is provided for the designer through the user interface.

6.2.2.2 FindValidN: micro-level KBE application

This subsection describes the knowledge of the KBE application FindValidN. The purpose of describing the knowledge behind the application is to show the reader the core function of the KBE system, which is to generate eMotor solution alternatives based only on the degrees of freedom of number of turns and length of stack. The objective called "Find eMotor Solution Alternatives for V-industrial" is created. The approach to accomplish the objective is pseudo-FEM computations in combination with Motorcad simulations. Since FEM simulations require high computational cost, the solution approach uses a database of FEM simulations comprising matrices from which interpolations are made to obtain a requested point. A tool to export the simulations in the format the KBE application needs it was developed and it is described in Appendix D. Using this approach enhances the searching of many solution alternatives in short computational time. In synthesis, each electric motor considered a reference for the Industrial V-cycle is fully characterized in FEM and the data file saved in the PDM prior its usage in the KBE application. In addition, for operating conditions the motor is simulated in Motorcad. The computational cost using Motorcad steady state capability is considered acceptable (fast enough), thus, the need of a database of losses (similar to FEM) is not required. This is also thought for further work expansion of the KBE application to the Component V-cycle.

As mentioned earlier this micro-level KBE applications make use of ModeladoScript. ModeladoScript is the application that uses the pseudo-FEM approach in order to compute the performance characteristics of a motor. It has a function called motorcadcalc, the application in charge of computing the change in temperature in Motorcad. For more information on ModeladoScript check Appendix D.

The main indications to consider a motor a solution alternative is shown in Fig. 6-4. The diagram says the following. The first step is to stablish the length of stack (Lstk) picked up from the inputs given in array form. If the Lstk is within range then the second step is to adjust the number of turns (N) to obtain rated torque within current limit. After this, validate the peak torque at ambient temperature condition by calculating the peak torque value and make sure the maximum voltage is not exceeded (at 90% of rated speed). If the peak torque is not fulfilled, go back and reduce the number of turns avoiding to surpass the current limit. If reducing the number of turns is not possible, increase the motor length.

Compute the motor losses at RMS torque. Once the losses are obtained, compute the copper and magnet temperatures at RMS torque operating condition. If the motor temperatures surpass the specified values increase the motor length. Then, validate the peak torque in operating temperature conditions by computing the torque and making sure the voltage limit is not exceeded. If the motor cannot reach the peak torque, go back and reduce the number of turns if this is not possible change the length of the electrical machine. Finally, if everything was within limits compute rated values at operating condition and the motor is considered a solution alternative. The indication to compute copper and magnet temperature change is allocated to the objective using the function Motorcadcalc. The rest shown in the diagram are allocated to the objective related with ModeladoScript.

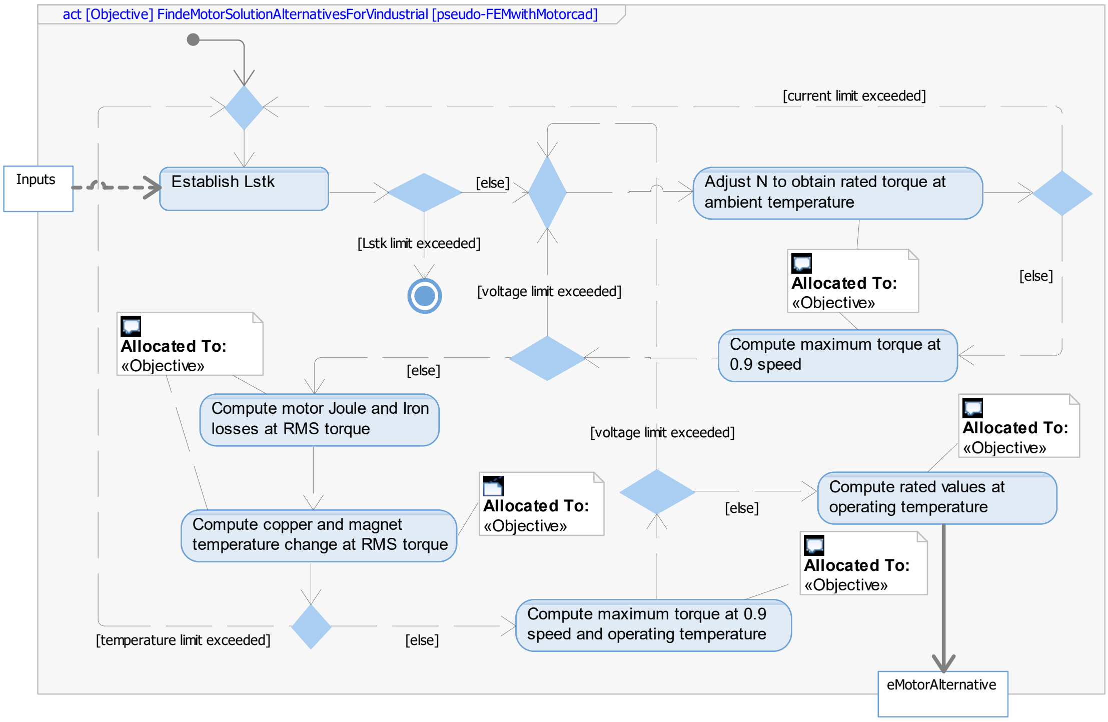


Fig. 6-4. Pseudo-FEM with Motorcad approach for generating eMotor alternatives

6.2.3 Package

Fig. 6-5 shows the architecture of the KBES with the selection of tools for its implementation. The KBE application is developed in Matlab App Designer because it offers the options of deploying the KBE application as a web application or a stand-alone application. Besides, the end-users are knowledgeable about coding in Matlab allowing adding features such as customized functions, for instance, in the developed SW the economic cost function has to be coded by the end-user in the same KBE application.

Even though it is technologically possible, by the time this thesis has been written, no middleware is developed for automatic integration among the PDM systems and Rhapsody with the KBE application. Thus, the process is done through manual operations. The drawback of manual operation is that if the metadata information in the PDM system is updated and the files are not exported or updated in the data repository, then the KBE application will work with old files leading to unwanted results. Similarly and regarding the MBSE tool the requirements values generated by the KBE application has to be entered manually not automatically, this could lead to mistakes if the process is not done properly. Despite the aforementioned issues, the KBES accomplish the objective of providing substantiation of the framework implementation.

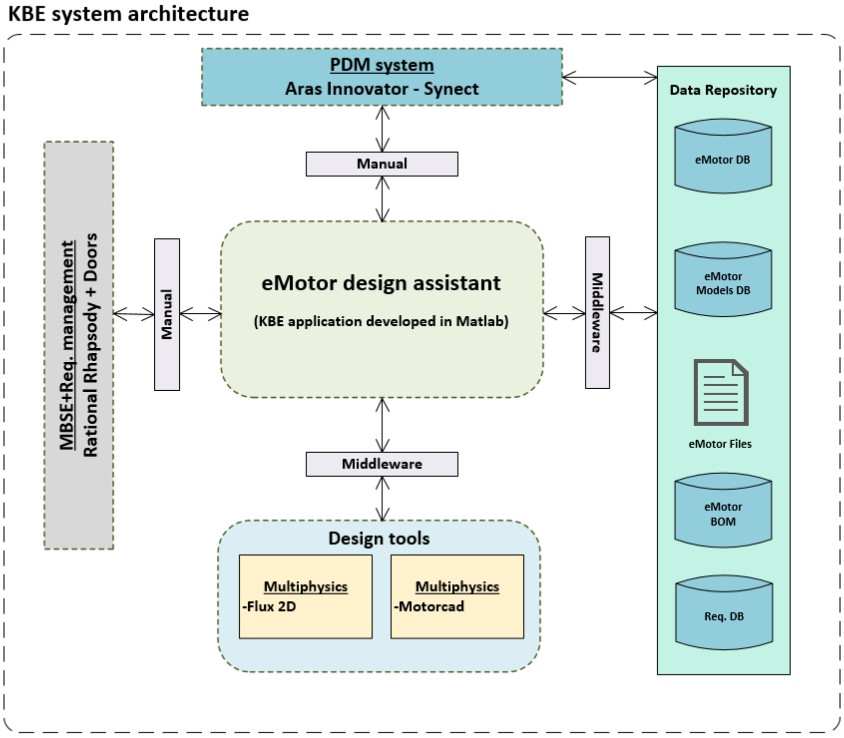
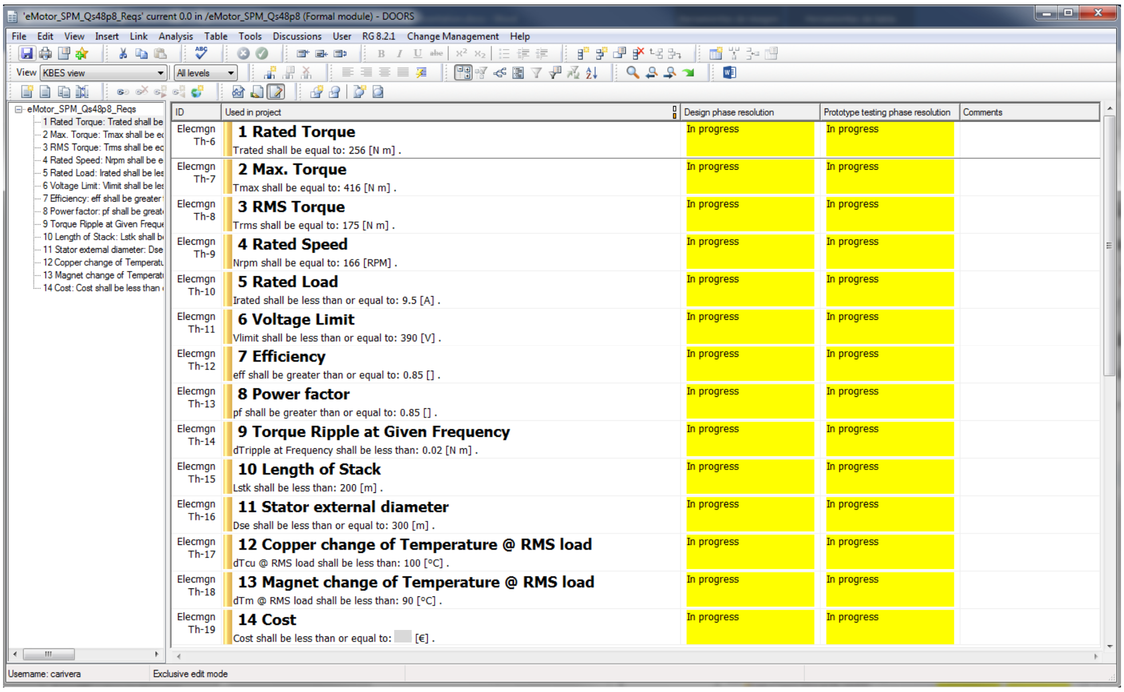


Fig. 6-5. KBE system implementation architecture.

The middleware to extract files or information from the data repository is mostly done through ActiveX technology of Microsoft. To integrate Motorcad with the KBE application a special function developed in VBscript is used as middleware and Python modules for Flux.

For every new project, prepared in advance templates for customization on each tool are provided to the user. Fig. 4-23 shows an example for Synect. Similarly, Fig. 6-6 shows the template in DOORS, the requirements specification is changed depending the project, however the requirements remain the same. The system is provided through a Rhapsody project file containing all the links among elements and models allowing the user to make few customizations like updating the elevator system parameters and requirements specifications Fig. 6-7a shows a satisfaction dependency between the requirements and motor parameters. If changes to requirements are made in either DOORS or Rhapsody these are automatically updated through the link in Gateway and shown in Fig. 6-7b. Regarding Aras Innovator, the customization of the metadata shown in Section 4.6 supports the user to stablish the properties to store and retrieve information about the structure of motors.



- eMotor_SPM_Qs48p8_Reqs
  - 1 Rated Torque: Trated shall be
  - 2 Max. Torque: Tmax shall be ec
  - 3 RMS Torque: Trms shall be ec
  - 4 Rated Speed: Nrpm shall be e
  - 5 Rated Load: Irated shall be les
  - 6 Voltage Limit: Vlimit shall be les
  - 7 Efficiency: eff shall be greater
  - 8 Power factor: pf shall be great
  - 9 Torque Ripple at Given Freque
  - 10 Length of Stack: Lstk shall b
  - 11 Stator external diameter: Dse
  - 12 Copper change of Temperatu
  - 13 Magnet change of Temperatu
  - 14 Cost: Cost shall be less than

ID	Used in project	Design phase resolution	Prototype testing phase resolution	Comments
Elecmgn Th-6	**1 Rated Torque** Trated shall be equal to: 256 [N m] .	In progress	In progress	
Elecmgn Th-7	**2 Max. Torque** Tmax shall be equal to: 416 [N m] .	In progress	In progress	
Elecmgn Th-8	**3 RMS Torque** Trms shall be equal to: 175 [N m] .	In progress	In progress	
Elecmgn Th-9	**4 Rated Speed** Nrpm shall be equal to: 166 [RPM] .	In progress	In progress	
Elecmgn Th-10	**5 Rated Load** Irated shall be less than or equal to: 9.5 [A] .	In progress	In progress	
Elecmgn Th-11	**6 Voltage Limit** Vlimit shall be less than or equal to: 390 [V] .	In progress	In progress	
Elecmgn Th-12	**7 Efficiency** eff shall be greater than or equal to: 0.85 [] .	In progress	In progress	
Elecmgn Th-13	**8 Power factor** pf shall be greater than or equal to: 0.85 [] .	In progress	In progress	
Elecmgn Th-14	**9 Torque Ripple at Given Frequency** dTripple at Frequency shall be less than: 0.02 [N m] .	In progress	In progress	
Elecmgn Th-15	**10 Length of Stack** Lstk shall be less than: 200 [m] .	In progress	In progress	
Elecmgn Th-16	**11 Stator external diameter** Dse shall be less than or equal to: 300 [m] .	In progress	In progress	
Elecmgn Th-17	**12 Copper change of Temperature @ RMS load** dTcu @ RMS load shall be less than: 100 [°C] .	In progress	In progress	
Elecmgn Th-18	**13 Magnet change of Temperature @ RMS load** dTm @ RMS load shall be less than: 90 [°C] .	In progress	In progress	
Elecmgn Th-19	**14 Cost** Cost shall be less than or equal to: [redacted] [€] .	In progress	In progress	

Username: carivera
Exclusive edit mode

Fig. 6-6. DOORS template for the KBE system for the development of eMotors for the elevator industry.

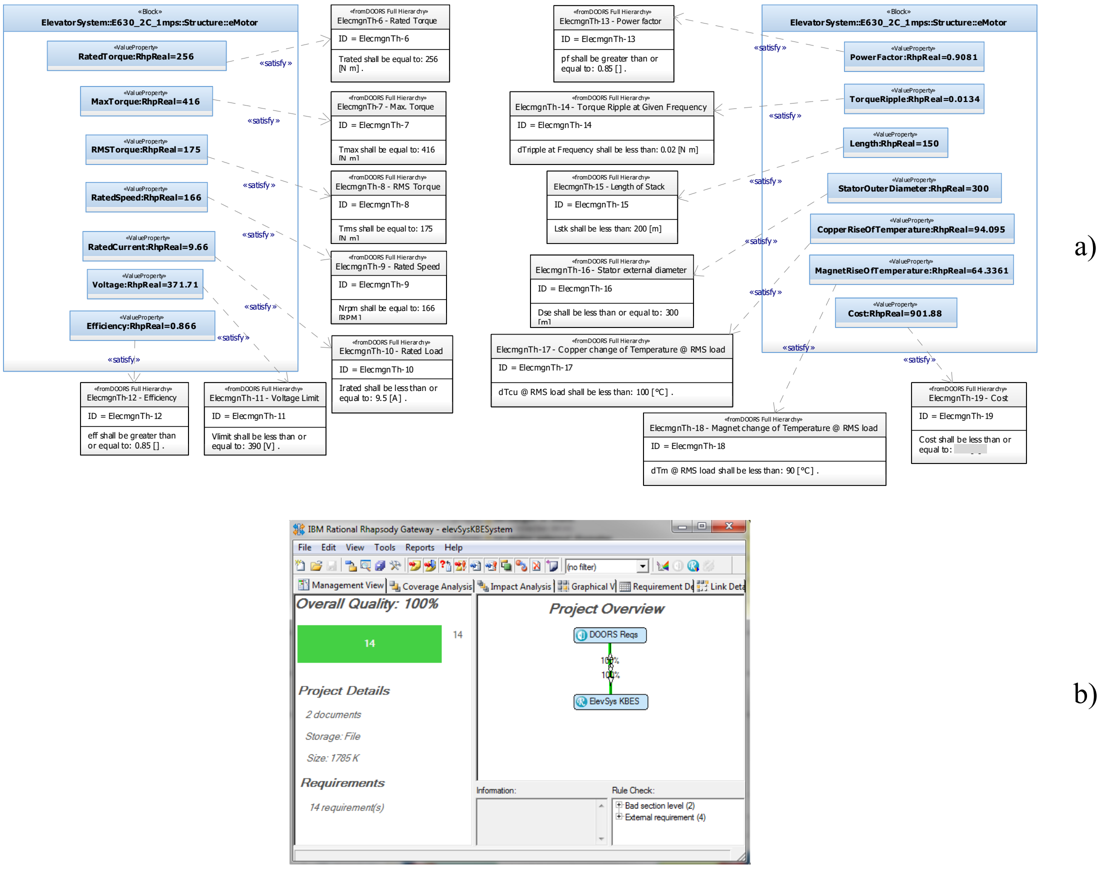


Fig. 6-7. a) Rhapsody requirements diagram; b) Gateway link between Rhapsody and DOORS.

The software of the KBE application is developed using modular programming to facilitate its maintenance and upgrading. Fig. 6-8 shows that for each stage one module is created, except for the application integration testing. The modules are prompted automatically according to the progress of the project.

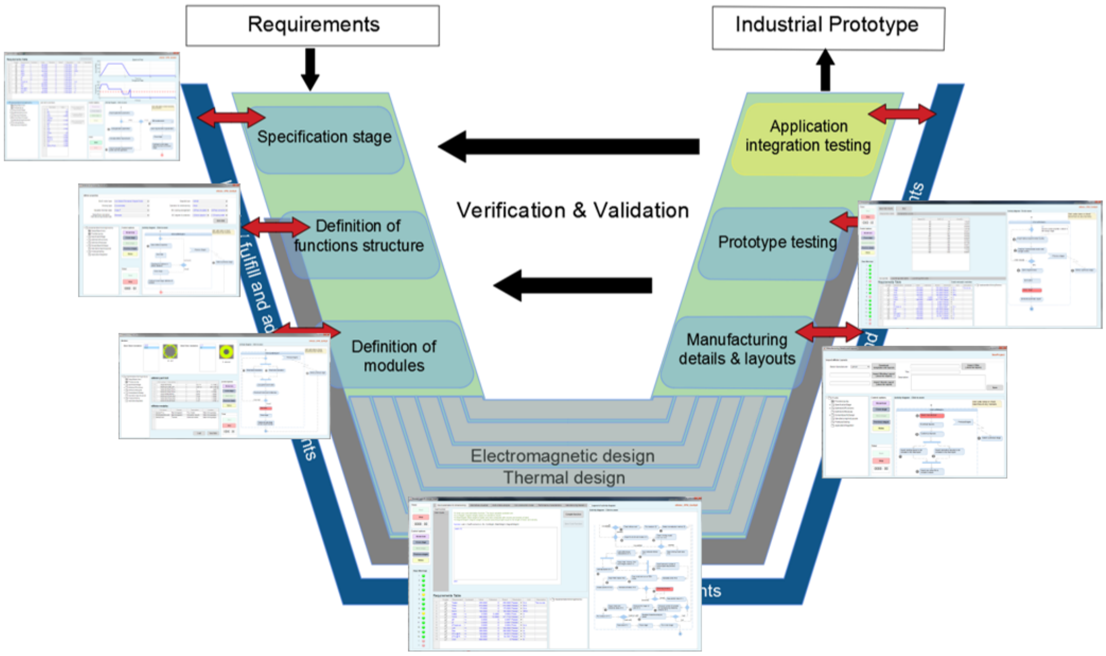
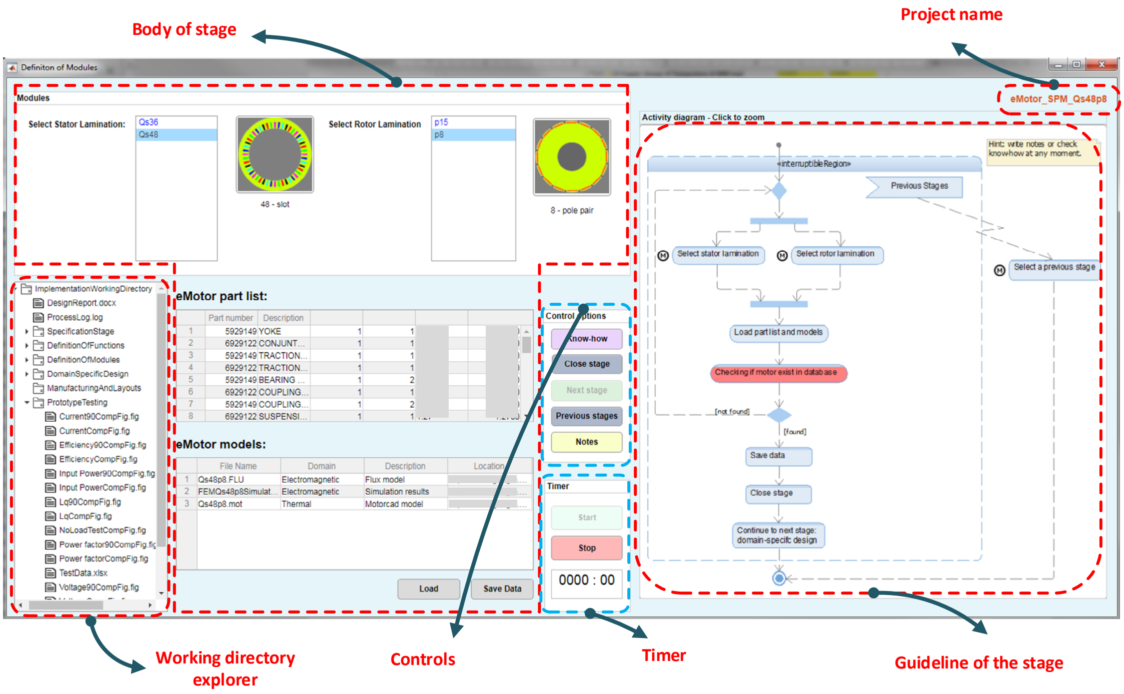


Fig. 6-8. Modular programming for the development of the KBE application.

The user interface for each module contains the same features. Fig. 6-9 illustrate these features, the only difference in each module is the body of the stage due to there is a particular purpose on each phase.



ImplementationWorkingDirectory
DesignReport.docx
ProcessLog.log
SpecificationStage
DefinitionOfFunctions
DefinitionOfModules
DomainSpecificDesign
ManufacturingAndLayouts
PrototypeTesting
Current90CompFig.fig
CurrentCompFig.fig
Efficiency90CompFig.fig
EfficiencyCompFig.fig
Input Power90CompFig.fig
Input PowerCompFig.fig
Lq90CompFig.fig
LqCompFig.fig
NoLoadTestCompFig.fig
Power factor90CompFig.fig
Power factorCompFig.fig
TestData.xlsx
Voltage90CompFig.fig

eMotor part list:

	Part number	Description				
1	5929149	YOKE	1	1	[redacted]	[redacted]
2	6929122	CONJUNT...	1	1	[redacted]	[redacted]
3	5929149	TRACTION...	1	1	[redacted]	[redacted]
4	6929122	TRACTION...	1	1	[redacted]	[redacted]
5	5929149	BEARING ...	1	2	[redacted]	[redacted]
6	6929122	COUPLING...	1	1	[redacted]	[redacted]
7	5929149	COUPLING...	1	2	[redacted]	[redacted]
8	6929122	SUSPENSI...	1	1	[redacted]	[redacted]

eMotor models:

	File Name	Domain	Description	Location
1	Qs48p8.FLU	Electromagnetic	Flux model	[redacted]
2	FEMQs48p8Simulat...	Electromagnetic	Simulation results	[redacted]
3	Qs48p8.mot	Thermal	Motorcad model	[redacted]

Fig. 6-9. Parts of the user interface in each module of eMotor Design Assistant.

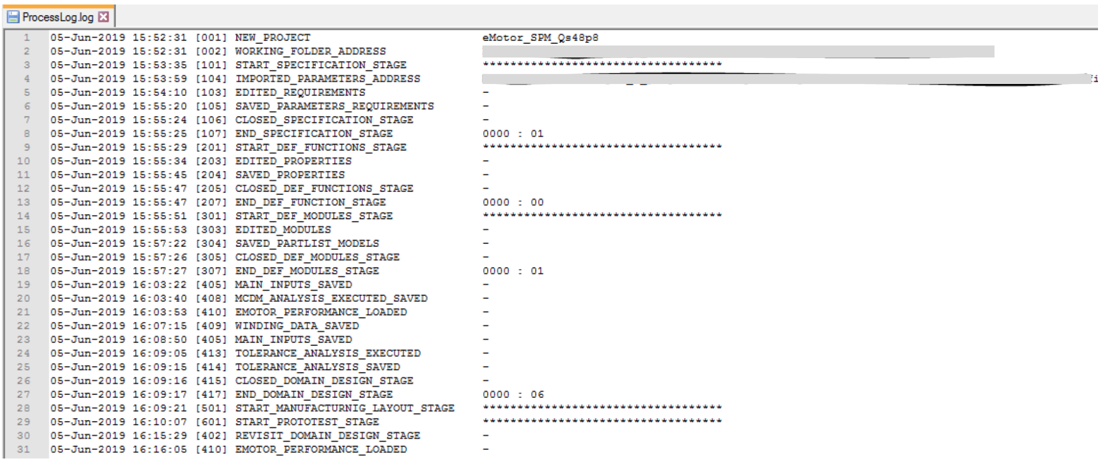
The KBE application offers the following advanced functionalities.

eMotor development process traceability. The KBE application tracks and trace the development process from start to end. The controls provide the possibility of going to a previous stage and close the current stage in order to proceed with the next stage.

Design requirements traceability and verification. Storage of parameters and values, results, and events in each stage. Capability to resume the project were it was left. Recording working time. In addition, the events on each stage are registered in a log file with a define structure available to the user at all times; see Fig. 6-10 for one example.

For every project, in the given path of the repository a folder for each stage is created where all the files corresponding to the stage will be exported automatically. The user can view the files through the working directory explorer in the KBE application. Furthermore, the KBE application has a timer that registered the time spent on each stage; the user can stop and start the timer in order to register only working time. Every time the stage is closed and later resumed the timer starts where it was left.

Finally, at the end of the process the KBE application generates a design report for the end-user with the information of each stage. For instance, selected properties and parameters, graphs, saved notes, motor solution with layouts, manufacturing details and a complete recapitulation of the development process with the statistics of time spent on each stage, revisits made to each stage and more.



```
05-Jun-2019 15:52:31 [001] NEW_PROJECT                         eMotor_SPM_Qs48p8
05-Jun-2019 15:52:31 [002] WORKING_FOLDER_ADDRESS              [redacted]
05-Jun-2019 15:53:35 [101] START_SPECIFICATION_STAGE           ***********************************
05-Jun-2019 15:53:59 [104] IMPORTED_PARAMETERS_ADDRESS         [redacted]
05-Jun-2019 15:54:10 [103] EDITED_REQUIREMENTS                 -
05-Jun-2019 15:55:20 [105] SAVED_PARAMETERS_REQUIREMENTS       -
05-Jun-2019 15:55:24 [106] CLOSED_SPECIFICATION_STAGE          -
05-Jun-2019 15:55:25 [107] END_SPECIFICATION_STAGE             0000 : 01
05-Jun-2019 15:55:29 [201] START_DEF_FUNCTIONS_STAGE           ***********************************
05-Jun-2019 15:55:34 [203] EDITED_PROPERTIES                   -
05-Jun-2019 15:55:45 [204] SAVED_PROPERTIES                    -
05-Jun-2019 15:55:47 [205] CLOSED_DEF_FUNCTIONS_STAGE          -
05-Jun-2019 15:55:47 [207] END_DEF_FUNCTION_STAGE              0000 : 00
05-Jun-2019 15:55:51 [301] START_DEF_MODULES_STAGE             ***********************************
05-Jun-2019 15:55:53 [303] EDITED_MODULES                      -
05-Jun-2019 15:57:22 [304] SAVED_PARTLIST_MODELS               -
05-Jun-2019 15:57:26 [305] CLOSED_DEF_MODULES_STAGE            -
05-Jun-2019 15:57:27 [307] END_DEF_MODULES_STAGE               0000 : 01
05-Jun-2019 16:03:22 [405] MAIN_INPUTS_SAVED                   -
05-Jun-2019 16:03:40 [408] MCDM_ANALYSIS_EXECUTED_SAVED        -
05-Jun-2019 16:03:53 [410] EMOTOR_PERFORMANCE_LOADED           -
05-Jun-2019 16:07:15 [409] WINDING_DATA_SAVED                  -
05-Jun-2019 16:08:50 [405] MAIN_INPUTS_SAVED                   -
05-Jun-2019 16:09:05 [413] TOLERANCE_ANALYSIS_EXECUTED         -
05-Jun-2019 16:09:15 [414] TOLERANCE_ANALYSIS_SAVED            -
05-Jun-2019 16:09:16 [415] CLOSED_DOMAIN_DESIGN_STAGE          -
05-Jun-2019 16:09:17 [417] END_DOMAIN_DESIGN_STAGE             0000 : 06
05-Jun-2019 16:09:21 [501] START_MANUFACTURNIG_LAYOUT_STAGE    ***********************************
05-Jun-2019 16:10:07 [601] START_PROTOTEST_STAGE               ***********************************
05-Jun-2019 16:15:29 [402] REVISIT_DOMAIN_DESIGN_STAGE         -
05-Jun-2019 16:16:05 [410] EMOTOR_PERFORMANCE_LOADED           -
```

Fig. 6-10. Log file: registration of events during the development proves by the KBE application.

Accessibility to knowledge. Retrieval of knowledge, information, and data. The KBE application integrates the knowledge saved in models with easy access to the user for context understanding. The controls seen in Fig. 6-9 give the possibility of accessing the know-how, which is knowledge the designer can read to understand in more detail the process. Each stage provides a guideline (see figure) for the user to finish successfully the phase. The guideline is an animated activity diagram remarking in red the current step of the stage for tracking and assurance all actions are made. Every time each stage is resumed the guideline shows the last executed action.

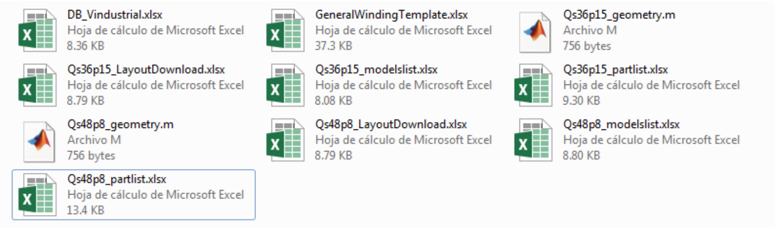
Automation of tasks. The KBE application automates complex tasks by creating intelligent algorithms as an outcome of the knowledge provided by the designer from the manual performance of tasks. Automatic requirements verification, automatic simulations performed in the design tools. Automatic cost analysis.

Some steps have to be done by the user through manual operation, the rest are executed automatically. The manual steps have an “M” symbol besides the action in the guideline, these steps typically requires judgement from the designer, therefore they are left as manual. The activity diagrams are also part of the micro-level knowledge representation of the KB.

Intelligent support to the designer. Solutions alternatives are obtained from validated models. Animated guideline to support the designer to get a valid solution. Different analysis tools were created to assist the designer in each step of the development process. Intelligent searching of solutions alternatives, warnings of requirements and variables out of range. Multi-criteria analysis tools, manufacturing tolerance analysis tool, error analysis comparison between design and prototype tests. The user can also create notes on each stage by clicking the notes button in the controls; the notes are saved in a text file and are included in the final design report.

6.3 KBE System implementation

This subsection presents step by step the implementation of the KBE system with emphasis on the KBE application remarking the process undertaken in order to create a viable motor solution. However, as mentioned earlier the files needed from the PDM systems are exported and stored manually in the repository (specific path) for accessibility to the KBE application. The files are the eMotor DB (containing the properties of each motor in the portfolio for the industrial V-cycle), the BOM, models list and layouts. For this particular case, two motors are available on the database and they are called the Qs36p15 and Qs48p8. The files are exported in Microsoft Excel format from Aras and Matlab format from Synect (except for models list), as future line of work the information withheld in the database of the PDM system should be accessed automatically by the KBE application to avoid this step. A screenshot of the files is shown in Fig. 6-11. Once this is done the KBE application can start running.


DB_Vindustrial.xlsx
Hoja de cálculo de Microsoft Excel
8.36 KB

GeneralWindingTemplate.xlsx
Hoja de cálculo de Microsoft Excel
37.3 KB

Qs36p15_geometry.m
Archivo M
756 bytes

Qs36p15_LayoutDownload.xlsx
Hoja de cálculo de Microsoft Excel
8.79 KB

Qs36p15_modelslist.xlsx
Hoja de cálculo de Microsoft Excel
8.08 KB

Qs36p15_partlist.xlsx
Hoja de cálculo de Microsoft Excel
9.30 KB

Qs48p8_geometry.m
Archivo M
756 bytes

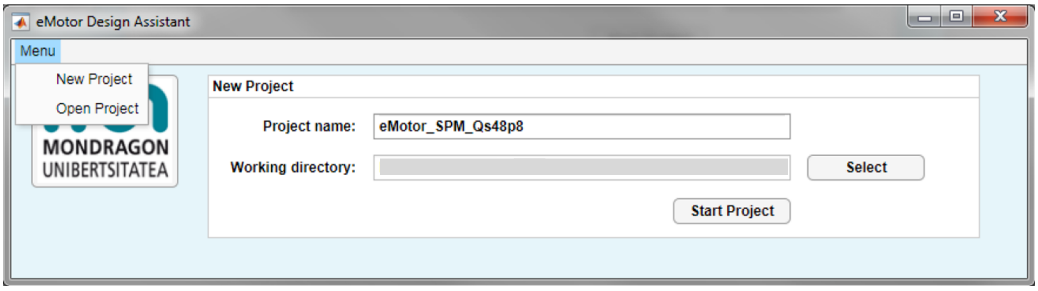
Qs48p8_LayoutDownload.xlsx
Hoja de cálculo de Microsoft Excel
8.79 KB

Qs48p8_modelslist.xlsx
Hoja de cálculo de Microsoft Excel
8.80 KB

Qs48p8_partlist.xlsx
Hoja de cálculo de Microsoft Excel
13.4 KB

Fig. 6-11. Exported files with database information from Aras and Synect.

An initial window appears where the name of the project and the path to the repository directory is entered. Fig. 6-12. illustrates one example, furthermore, when the open option is selected from the menu the project is resumed in the last stage and step the end-user was working on. Each stage of the KBE system will be explained from the KBE application perspective and its relationship with the designer and the rest of product development technologies mentioned throughout this work.



Project name: eMotor_SPM_Qs48p8

Working directory:

Fig. 6-12. Initial window of eMotor Design Assistant.

6.3.1 Specification stage

The elevator system and its parameters needed by the KBE application are shown in Fig. 6-13. It can be seen from the figure the use of SysML to specify the elevator system with the parameters of interest for the development of the motor. The user should update the values in Rhapsody as shown in the figure in order to have consistency among the tools. These parameters are incorporated in the prepared in advanced template of Synect; the values can be updated and exported in Matlab format to the repository. Fig. 6-14 gives an illustration of the process.

Moreover, Fig. 6-15 shows the specification stage of the KBE application. The aforementioned elevator parameters are imported and automatically displayed in the user interface. The designer must follow the guideline to finish the process successfully. Some manual steps can be skipped if its execution is not needed, for instance, in this example the values of the elevator parameters requires no edition since they were exported with the correct values from Synect. However, the option is still available for the designer to edit the parameters and compute the requirements for analysis.

The automatically computed requirements are the rated torque (Trated), peak torque (Tmax), RMS torque (Trms) and speed in RPM. Furthermore, the characteristic curves are generated automatically. Nevertheless, the designer must fill the requirement constraints and values for the rest of requirements, if the requirement enable field is disable that requirement is not taken into account along the entire process. In addition, the KBE application offers the capability of adding two additional customized requirements (add requirements button). However, if this feature is used these requirements must be added manually in DOORS and synchronize it with Rhapsody too.

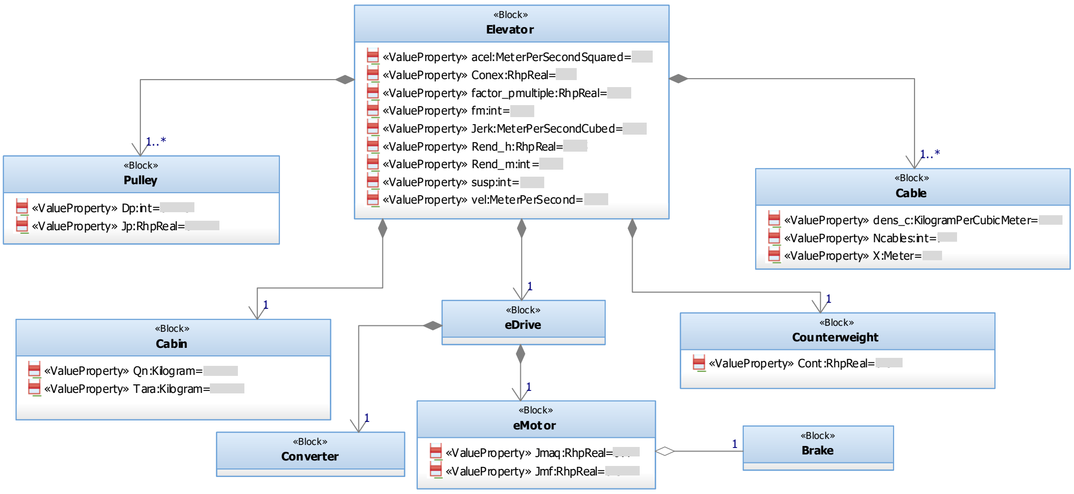
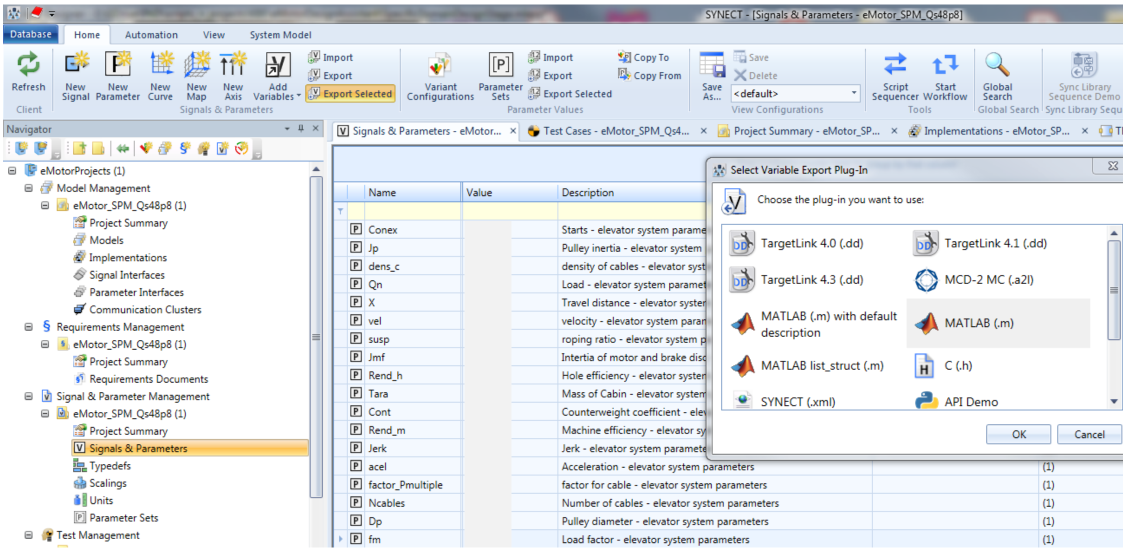


Fig. 6-13. Elevator system with parts and parameters.

Finally, the outcome of this phase are the files shown in the working directory explorer in Fig. 6-15. In case the end-user needs to return to this phase in the future, the KBE application will reload the data from these files and keep track of the process in the background.



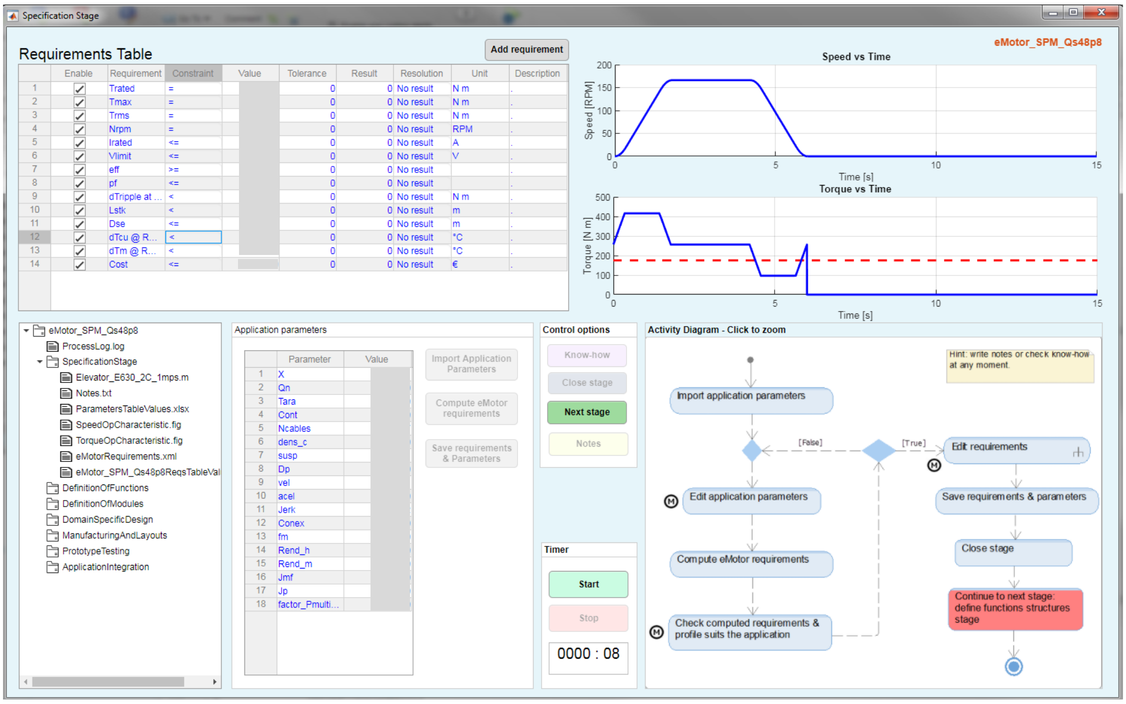
	Name	Value	Description	
P	Conex		Starts - elevator system parame	
P	Jp		Pulley inertia - elevator system	
P	dens_c		density of cables - elevator syst	
P	Qn		Load - elevator system paramet	
P	X		Travel distance - elevator syster	
P	vel		velocity - elevator system paran	
P	susp		roping ratio - elevator system p	
P	Jmf		Intertia of motor and brake disc	
P	Rend_h		Hole efficiency - elevator syster	
P	Tara		Mass of Cabin - elevator system	
P	Cont		Counterweight coefficient - elev	
P	Rend_m		Machine efficiency - elevator sy	
P	Jerk		Jerk - elevator system paramete	
P	acel		Acceleration - elevator system parameters	(1)
P	factor_Pmultiple		factor for cable - elevator system parameters	(1)
P	Ncables		Number of cables - elevator system parameters	(1)
P	Dp		Pulley diameter - elevator system parameters	(1)
P	fm		Load factor - elevator system parameters	(1)

Select Variable Export Plug-In

Choose the plug-in you want to use:

- TargetLink 4.0 (.dd)
- TargetLink 4.1 (.dd)
- TargetLink 4.3 (.dd)
- MCD-2 MC (.a2l)
- MATLAB (.m) with default description
- MATLAB (.m)
- MATLAB list_struct (.m)
- C (.h)
- SYNECT (.xml)
- API Demo

Fig. 6-14. Exporting elevator parameters with values for specification stage.



Requirements Table

	Enable	Requirement	Constraint	Value	Tolerance	Result	Resolution	Unit	Description
1	☑	Trated	=		0	0	No result	N m	.
2	☑	Tmax	=		0	0	No result	N m	.
3	☑	Trms	=		0	0	No result	N m	.
4	☑	Nrpm	=		0	0	No result	RPM	.
5	☑	Irated	<=		0	0	No result	A	.
6	☑	Vlimit	<=		0	0	No result	V	.
7	☑	eff	>=		0	0	No result		.
8	☑	pf	<=		0	0	No result		.
9	☑	dTripple at ...	<		0	0	No result	N m	.
10	☑	Lstk	<		0	0	No result	m	.
11	☑	Dse	<=		0	0	No result	m	.
12	☑	dTcu @ R...	<		0	0	No result	°C	.
13	☑	dTm @ R...	<		0	0	No result	°C	.
14	☑	Cost	<=		0	0	No result	€	.

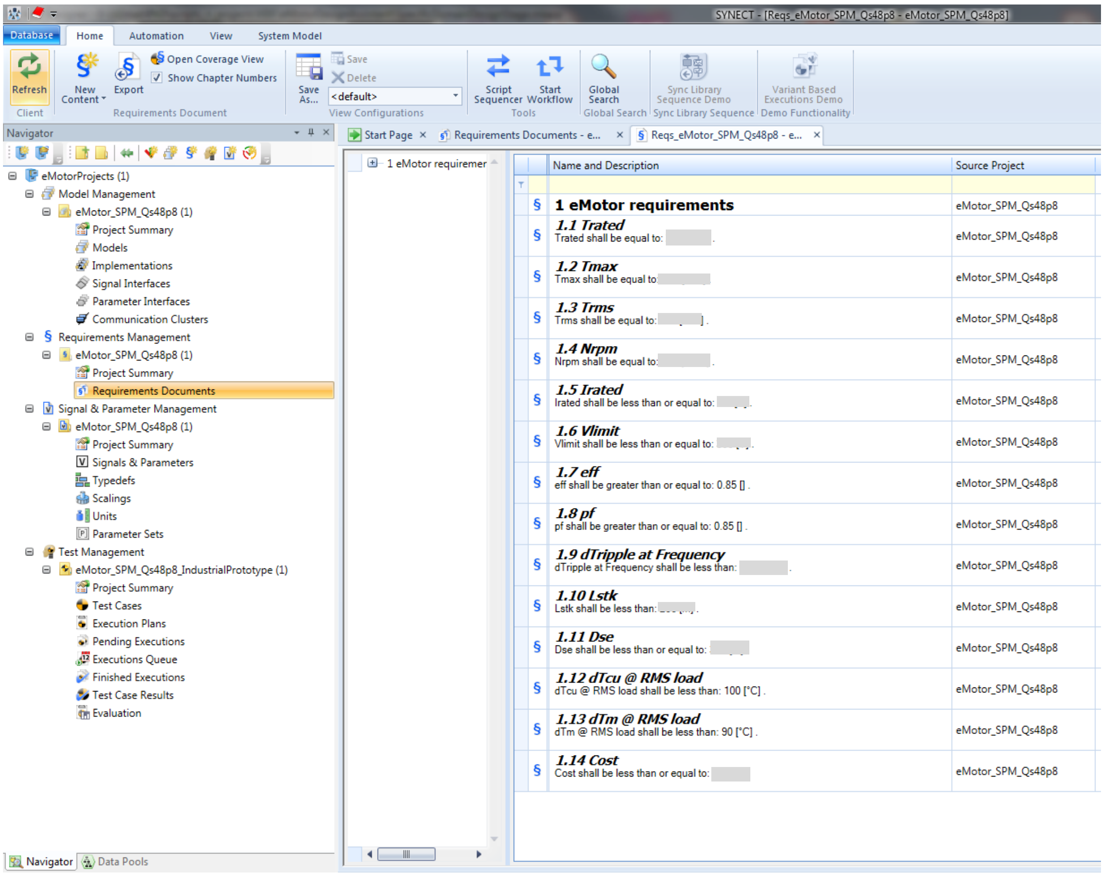
- eMotor_SPM_Qs48p8
  - ProcessLog.log
  - SpecificationStage
    - Elevator_E630_2C_1mps.m
    - Notes.txt
    - ParametersTableValues.xlsx
    - SpeedOpCharacteristic.fig
    - TorqueOpCharacteristic.fig
    - eMotorRequirements.xml
    - eMotor_SPM_Qs48p8ReqsTableVal
  - DefinitionOfFunctions
  - DefinitionOfModules
  - DomainSpecificDesign
  - ManufacturingAndLayouts
  - PrototypeTesting
  - ApplicationIntegration

Application parameters

	Parameter	Value
1	X	
2	Qn	
3	Tara	
4	Cont	
5	Ncables	
6	dens_c	
7	susp	
8	Dp	
9	vel	
10	acel	
11	Jerk	
12	Conex	
13	fm	
14	Rend_h	
15	Rend_m	
16	Jmf	
17	Jp	
18	factor_Pmulti...	

Fig. 6-15. User interface of the specification stage in eMotor Design Assistant (KBE application).

Following the KBE system actions shown in Fig. 6-3 we need to load the low level requirements into Synect and change the requirements specification in Rhapsody and synchronize it with DOORS. The KBE application generates automatically an XML file with the Synect requirements structure with the requirements specifications ready to be loaded, see Fig. 6-16. Regarding Rhapsody and DOORS Fig. 6-6 and Fig. 6-7 shows the aforementioned process.



	Name and Description	Source Project
§	**1 eMotor requirements**	eMotor_SPM_Qs48p8
§	***1.1 Trated*** Trated shall be equal to: [redacted] .	eMotor_SPM_Qs48p8
§	***1.2 Tmax*** Tmax shall be equal to [redacted]	eMotor_SPM_Qs48p8
§	***1.3 Trms*** Trms shall be equal to: [redacted]] .	eMotor_SPM_Qs48p8
§	***1.4 Nrpm*** Nrpm shall be equal to: [redacted] .	eMotor_SPM_Qs48p8
§	***1.5 Irated*** Irated shall be less than or equal to: [redacted].	eMotor_SPM_Qs48p8
§	***1.6 Vlimit*** Vlimit shall be less than or equal to: [redacted].	eMotor_SPM_Qs48p8
§	***1.7 eff*** eff shall be greater than or equal to: 0.85 [] .	eMotor_SPM_Qs48p8
§	***1.8 pf*** pf shall be greater than or equal to: 0.85 [] .	eMotor_SPM_Qs48p8
§	***1.9 dTripple at Frequency*** dTripple at Frequency shall be less than: [redacted] .	eMotor_SPM_Qs48p8
§	***1.10 Lstk*** Lstk shall be less than: [redacted] .	eMotor_SPM_Qs48p8
§	***1.11 Dse*** Dse shall be less than or equal to: [redacted]	eMotor_SPM_Qs48p8
§	***1.12 dTcu @ RMS load*** dTcu @ RMS load shall be less than: 100 [°C] .	eMotor_SPM_Qs48p8
§	***1.13 dTm @ RMS load*** dTm @ RMS load shall be less than: 90 [°C] .	eMotor_SPM_Qs48p8
§	***1.14 Cost*** Cost shall be less than or equal to: [redacted]	eMotor_SPM_Qs48p8

Fig. 6-16. Requirements file generated by the KBE application and loaded in Synect for test case management.

The guideline for the user in this phase is illustrated in Fig. 6-17.

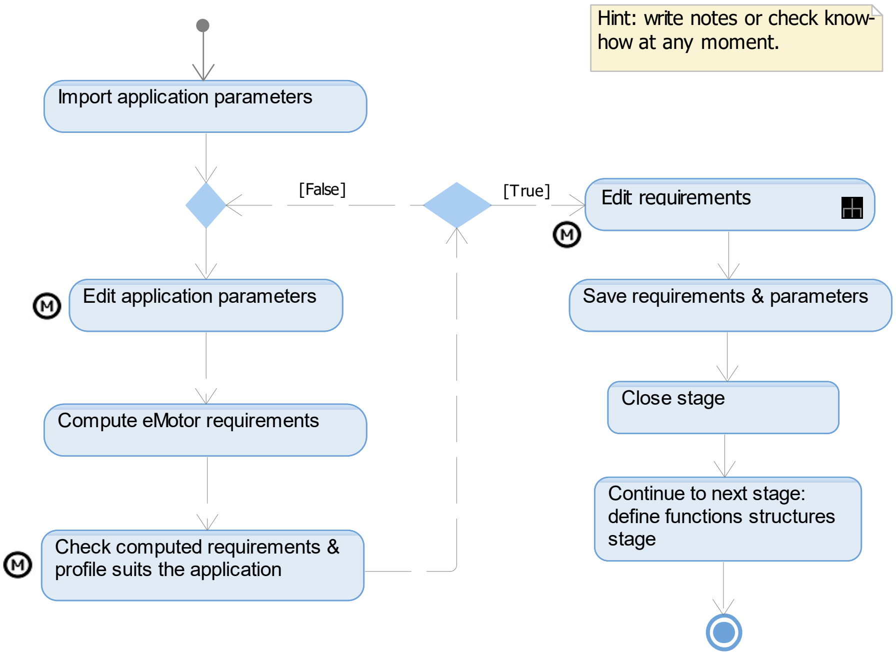
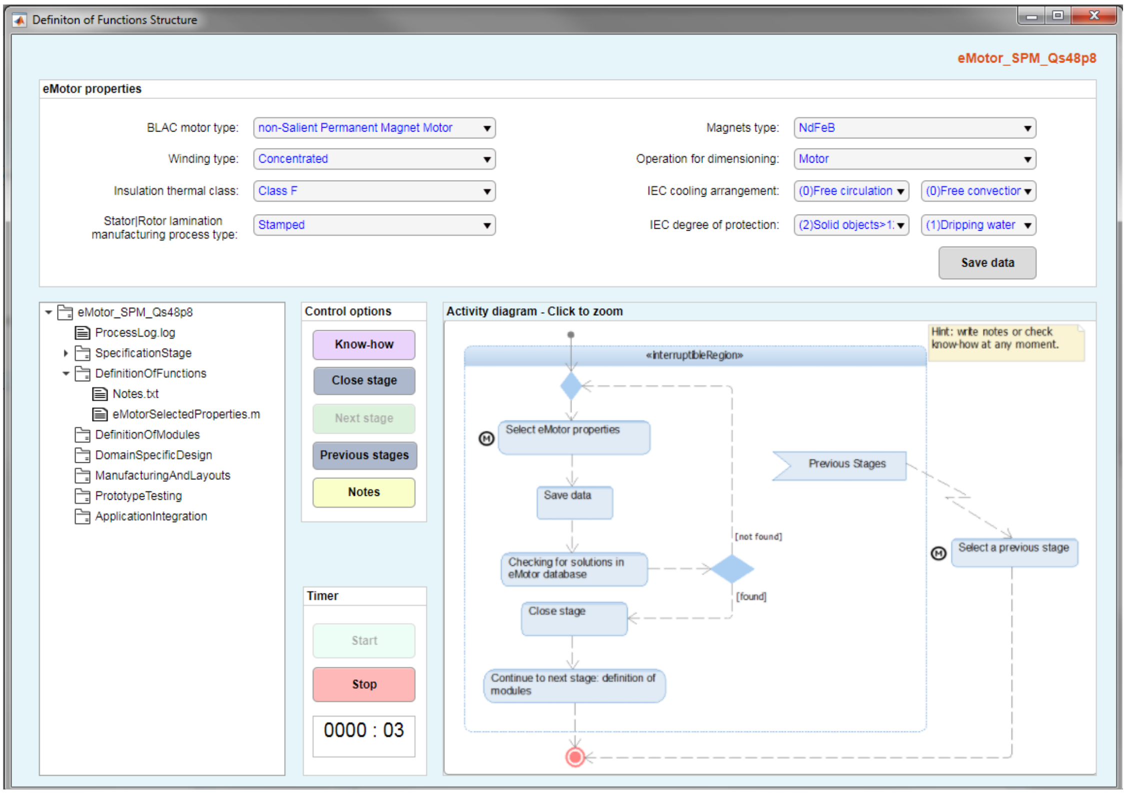


Fig. 6-17. Guideline embedded in the KBE application for the specification stage.

6.3.2 Definition of functions structure

In this stage, the goal is to limit the solutions space. The designer limits the space of solution by the selection of properties of the motor. In this example, only two motors are available, nonetheless, hoping the reader can foresee its importance when many motors are stored in the database. Fig. 6-18 shows the user interface for this stage. The designer selects the properties (same properties imported from Aras), once the data is saved the application checks for solutions in the database and creates a file for the next stage in the background (not available to the end-user) with the motors that can become a possible solution. The only output here is a file with the selected properties to serve as a reminder to the designer for replicability (see the directory explorer in the figure).

Following the KBE system activity diagram (Fig. 6-3) and recalling the files containing the properties, part lists and models list were prepared beforehand in order to execute the KBE application (Fig. 6-11). The KBE application continues by filtering from the aforementioned files the motors that fulfills the selected properties for their loading in the next stage. However, for future line of work the filtering should be done automatically from the PDM system database, same comment applies for the next stage.


Definiton of Functions Structure

eMotor_SPM_Qs48p8

eMotor properties

BLAC motor type: non-Salient Permanent Magnet Motor
Winding type: Concentrated
Insulation thermal class: Class F
Stator|Rotor lamination manufacturing process type: Stamped
Magnets type: NdFeB
Operation for dimensioning: Motor
IEC cooling arrangement: (0)Free circulation (0)Free convectior
IEC degree of protection: (2)Solid objects>1: (1)Dripping water

- eMotor_SPM_Qs48p8
  - ProcessLog.log
  - SpecificationStage
  - DefinitionOfFunctions
    - Notes.txt
    - eMotorSelectedProperties.m
  - DefinitionOfModules
  - DomainSpecificDesign
  - ManufacturingAndLayouts
  - PrototypeTesting
  - ApplicationIntegration

Timer

0000 : 03

Hint: write notes or check know-how at any moment.

Fig. 6-18. User interface of the definition of functions structure stage.

The animated guideline embedded in the KBE application for the user in this phase is presented in Fig. 6-19.

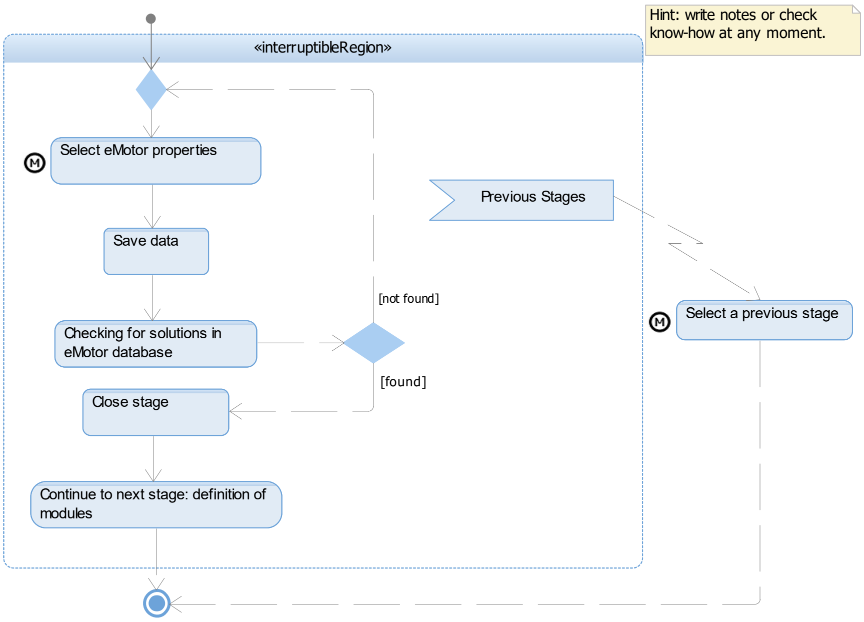


Fig. 6-19. Guideline embedded in the KBE application for the definition of functions structure stage.

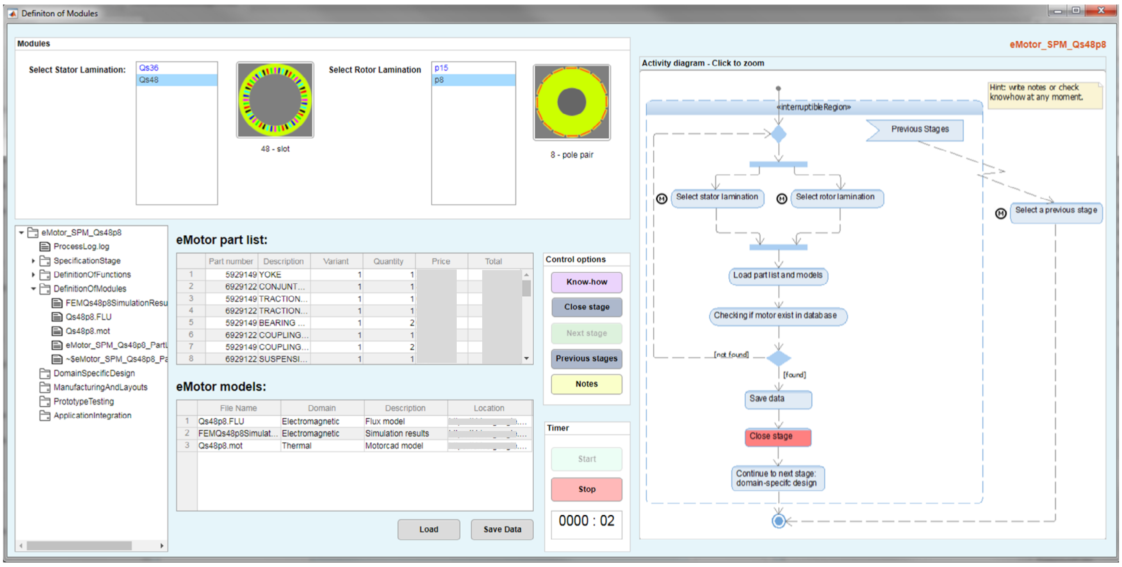
6.3.3 Definition of modules

In this stage, the designer selects the stator and rotor lamination from the ones that appear available in each list, check Fig. 6-20. The stator and rotor lamination available depend on the possible solutions found in the previous stage based on the properties selected by the designer.

The combination between stator and rotor lamination is checked upon the motor solutions, i.e. if the stator and rotor combination does not exist, the designer must select a new combination. After the selection is made, the designer loads the parts and models list and saves them by pressing the load and save buttons respectively. The KBE application downloads automatically the files using the links provided in the models list location field and stores them in the repository.

The output in this stage are the files containing these part lists and the electromagnetic and thermal models as well as the FEM simulation results in XML format (or xlsx) for pseudo-FEM simulations mentioned earlier in this chapter.

The animated guideline embedded in the KBE application for the end-user in this phase is shown in Fig. 6-21.



eMotor part list:

	Part number	Description	Variant	Quantity	Price	Total
1	5929149	YOKE	1	1	[redacted]	[redacted]
2	6929122	CONJUNT...	1	1	[redacted]	[redacted]
3	5929149	TRACTION...	1	1	[redacted]	[redacted]
4	6929122	TRACTION...	1	1	[redacted]	[redacted]
5	5929149	BEARING ...	1	2	[redacted]	[redacted]
6	6929122	COUPLING...	1	1	[redacted]	[redacted]
7	5929149	COUPLING...	1	2	[redacted]	[redacted]
8	6929122	SUSPENSI...	1	1	[redacted]	[redacted]

eMotor models:

	File Name	Domain	Description	Location
1	Qs48p8.FLU	Electromagnetic	Flux model	[redacted]
2	FEMQs48p8Simulat...	Electromagnetic	Simulation results	[redacted]
3	Qs48p8.mot	Thermal	Motorcad model	[redacted]

0000 : 02

Fig. 6-20. User interface for the definition of modules stage.

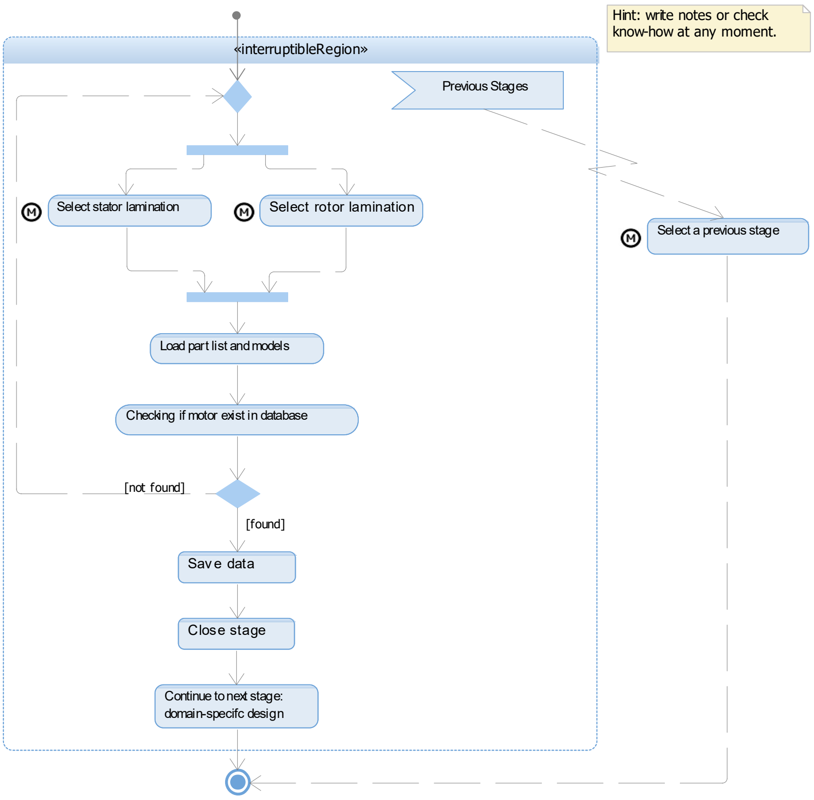


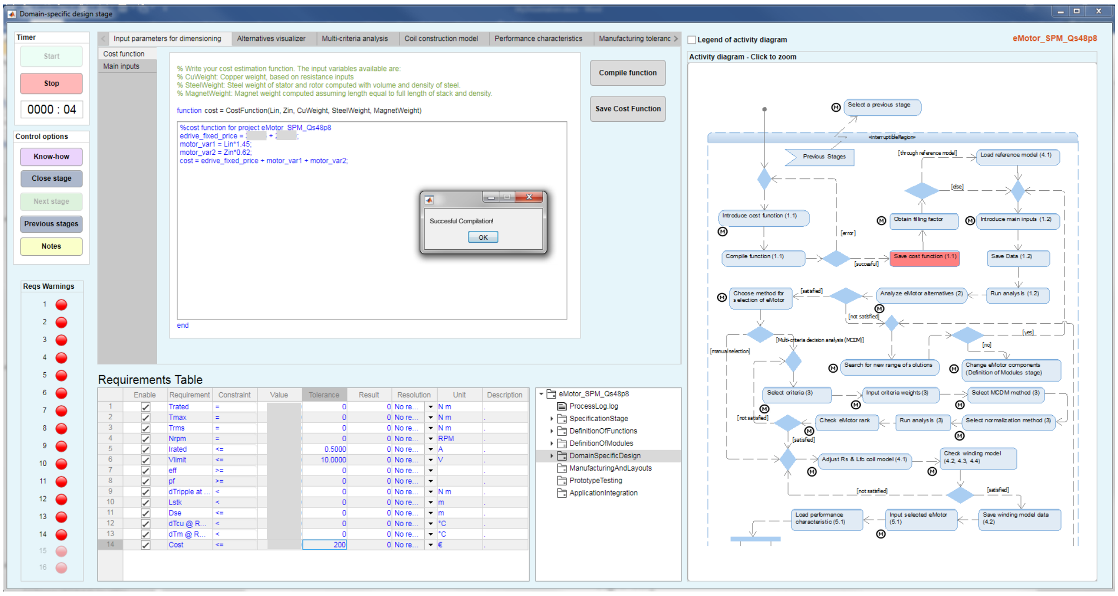
Fig. 6-21. Guideline embedded in the KBE application for the definition of modules stage.

6.3.4 Domain-specific design stage

The first step in this stage is to code the cost function. The designer has available as input variables the length, number of turns, copper weight, steel weight and magnet weight to be used in the cost functions. The cost function is coded using Matlab commands, in the example shown in Fig. 6-22 only the length and number of turns are used. The input cost function is also reported in the final report appearing as the following example.

```
function cost = CostFunction(Lin, Zin, CuWeight, SteelWeight, MagnetWeight)
---- %cost function for project eMotor_SPM_Qs48p8
---- edrive_fixed_price = C1+ C2;
---- motor_var1 = Lin*1.45;
---- motor_var2 = Zin*0.62;
---- cost = edrive_fixed_price + motor_var1 + motor_var2;
end
```

There is a compile function button that the user must click in order to check if the entered code is correct as shown in the figure, then the cost function is saved.



```
% Write your cost estimation function. The input variables available are:
% CuWeight: Copper weight, based on resistance inputs
% SteelWeight: Steel weight of stator and rotor computed with volume and density of steel.
% MagnetWeight: Magnet weight computed assuming length equal to full length of stack and density.

function cost = CostFunction(Lin, Zin, CuWeight, SteelWeight, MagnetWeight)
%cost function for project eMotor_SPM_Qs48p8
edrive_fixed_price = [redacted] + [redacted];
motor_var1 = Lin*1.45;
motor_var2 = Zin*0.62;
cost = edrive_fixed_price + motor_var1 + motor_var2;
end
```

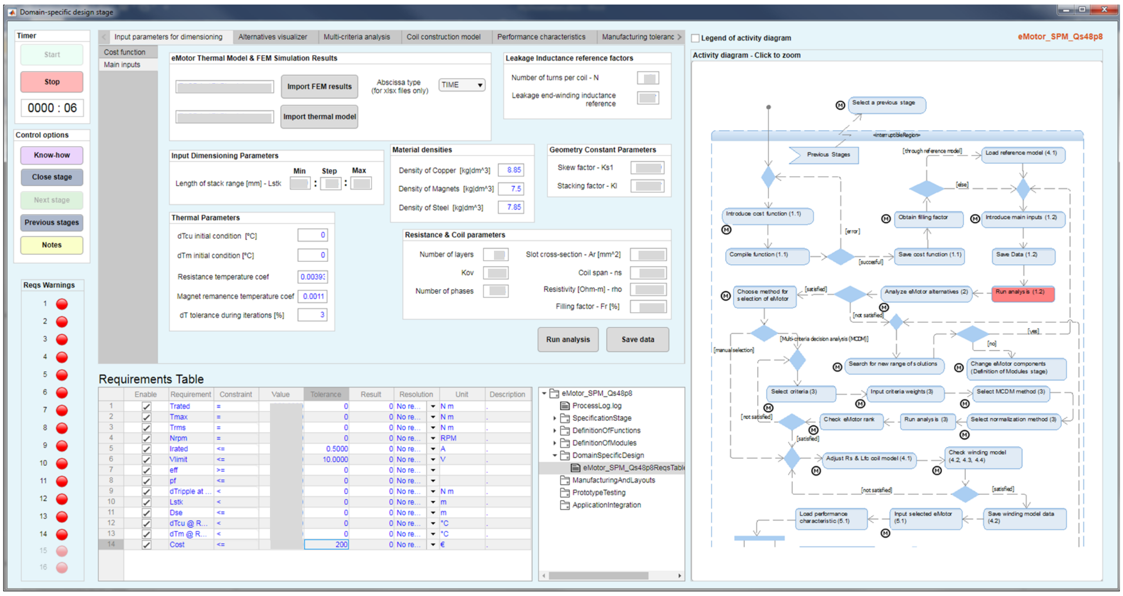
Succesful Compilation!

Requirements Table

	Enable	Requirement	Constraint	Value	Tolerance	Result	Resolution	Unit	Description
1	☑	Trated	=	[redacted]	0	0	No re...	N m	.
2	☑	Tmax	=	[redacted]	0	0	No re...	N m	.
3	☑	Trms	=	[redacted]	0	0	No re...	N m	.
4	☑	Nrpm	=	[redacted]	0	0	No re...	RPM	.
5	☑	Irated	<=	[redacted]	0.5000	0	No re...	A	.
6	☑	Vlimit	<=	[redacted]	10.0000	0	No re...	V	.
7	☑	eff	>=	[redacted]	0	0	No re...		.
8	☑	pf	>=	[redacted]	0	0	No re...		.
9	☑	dTripple at ...	<	[redacted]	0	0	No re...	N m	.
10	☑	Lstk	<	[redacted]	0	0	No re...	m	.
11	☑	Dse	<=	[redacted]	0	0	No re...	m	.
12	☑	dTcu @ R...	<	[redacted]	0	0	No re...	°C	.
13	☑	dTm @ R...	<	[redacted]	0	0	No re...	°C	.
14	☑	Cost	<=	[redacted]	200	0	No re...	€	.

Fig. 6-22. Domain-specific design stage: Input parameters for dimensioning tab, cost function.

The next step is introduced in Fig. 6-23. It consist of entering the inputs (main inputs tab), most inputs are automatically loaded from the reference model of the motor selected in the previous stage. The user should only enter the filling factor, the Lstk range, the dT tolerance (this is the maximum error accepted with thermal simulation iterations to stop the process) and the files containing the FEM results and thermal model saved in the folder from the previous step. In this example the Lstk has been set from 1 pu to 1.75 pu with a step of 0.25 pu because the magnets available in stock have these sizes. In addition, in the requirements table the user can insert tolerance values, the tolerance values allow the user to see solutions outside the requirements constraints. In this example, solutions with a current surpassing up to 5.26% the given limit are considered as well as a voltage limit up to 400 Volts and the economic cost can reach up to 22% the target cost. Then, the designer can click the run analysis button; behind the scene the micro-level KBE application FindValidN (explained in Section 6.2.2.2) is executed. Depending the range of Lstk entered this process takes a while to finish, for the inputs mentioned the process finished in three minutes.



0000 : 06

eMotor Thermal Model & FEM Simulation Results

Abscissa type (for xlsx files only): TIME

Leakage Inductance reference factors

Number of turns per coil - N

Leakage end-winding inductance reference

Input Dimensioning Parameters

Length of stack range [mm] - Lstk: Min : Step : Max

Material densities

Density of Copper [kg|dm^3]: 8.85

Density of Magnets [kg|dm^3]: 7.5

Density of Steel [kg|dm^3]: 7.85

Geometry Constant Parameters

Skew factor - Ks1

Stacking factor - Kl

Thermal Parameters

dTcu initial condition [°C]: 0

dTm initial condition [°C]: 0

Resistance temperature coef: 0.0039:

Magnet remanence temperature coef: 0.0011

dT tolerance during iterations [%]: 3

Resistance & Coil parameters

Number of layers

Kov

Number of phases

Slot cross-section - Ar [mm^2]

Coil span - ns

Resistivity [Ohm-m] - rho

Filling factor - Fr [%]

Requirements Table

	Enable	Requirement	Constraint	Value	Tolerance	Result	Resolution	Unit	Description
1	☑	Trated	=	[redacted]	0	0	No re...	N m	.
2	☑	Tmax	=	[redacted]	0	0	No re...	N m	.
3	☑	Trms	=	[redacted]	0	0	No re...	N m	.
4	☑	Nrpm	=	[redacted]	0	0	No re...	RPM	.
5	☑	Irated	<=	[redacted]	0.5000	0	No re...	A	.
6	☑	Vlimit	<=	[redacted]	10.0000	0	No re...	V	.
7	☑	eff	>=	[redacted]	0	0	No re...		.
8	☑	pf	<=	[redacted]	0	0	No re...		.
9	☑	dTripple at ...	<	[redacted]	0	0	No re...	N m	.
10	☑	Lstk	<	[redacted]	0	0	No re...	m	.
11	☑	Dse	<=	[redacted]	0	0	No re...	m	.
12	☑	dTcu @ R...	<	[redacted]	0	0	No re...	°C	.
13	☑	dTm @ R...	<	[redacted]	0	0	No re...	°C	.
14	☑	Cost	<=	[redacted]	200	0	No re...	€	.

- eMotor_SPM_Qs48p8
  - ProcessLog.log
  - SpecificationStage
  - DefinitionOfFunctions
  - DefinitionOfModules
  - DomainSpecificDesign
    - eMotor_SPM_Qs48p8ReqsTabl
  - ManufacturingAndLayouts
  - PrototypeTesting
  - ApplicationIntegration

Fig. 6-23. Domain-specific design stage: Input parameters for dimensioning tab, main inputs.

After the solution alternatives are generated, the KBE application provides the capability of checking the results in the same software. The entered inputs generated eight motor alternatives two for the length of 1.5 pu and six for the length of 1.75 pu, no solutions were generated for Lstk equal to 1 pu. Fig. 6-24 shows the tools available for the analysis of these solutions, the KBE application allows the capability of creating bars graph for any selected parameters, it provides a table of cost, weight, and volume as well as performance tables showing the data for both ambient and operating temperature for each motor alternative.

Moreover, the following step is to select a motor from all the alternatives. The KBE application provides four MCDM methods for the task. The methods are TOPSIS, WSM, WPM and WASPAS for references please check [134], [135]. The designer must select the criteria (nine available) and input their weights according to his judgement. Fig. 6-25 shows a criteria combination example using TOPSIS giving as best alternative the motor (L1.5 pu, N2.04 pu). In charge of the process is the micro-level KBE application MCDMAnalysis.

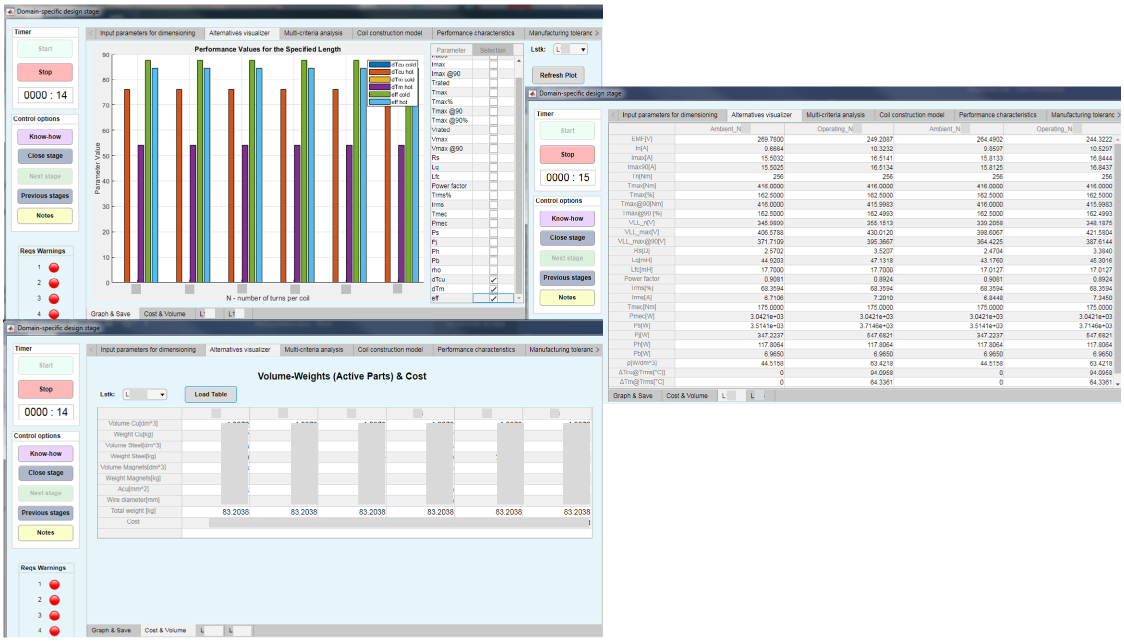


Fig. 6-24. Domain-specific design stage: Alternatives visualizer tab, tools for analysis.

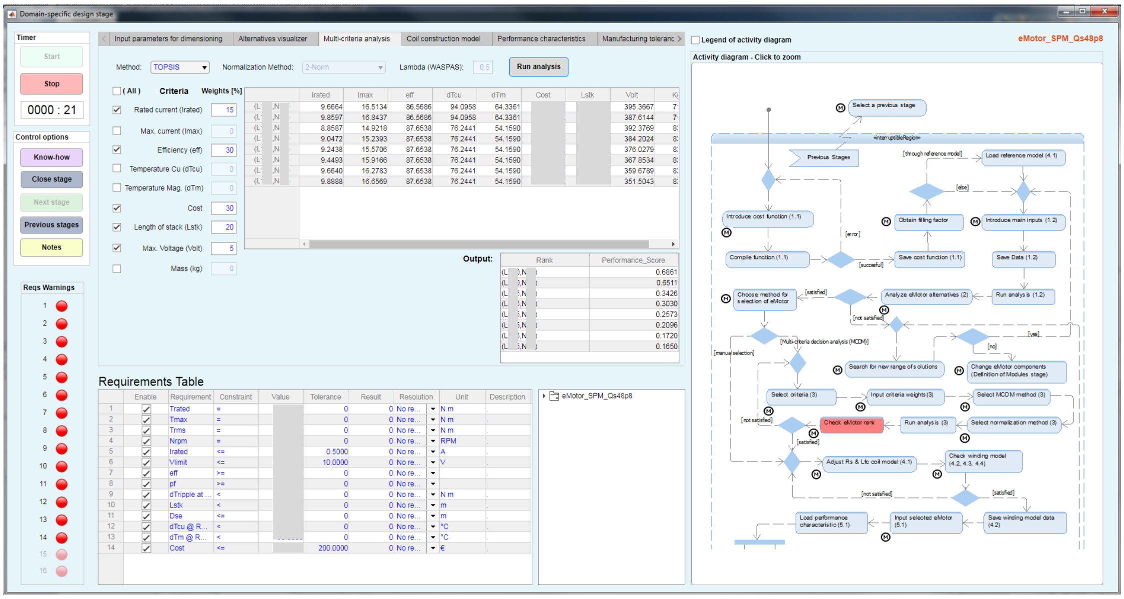
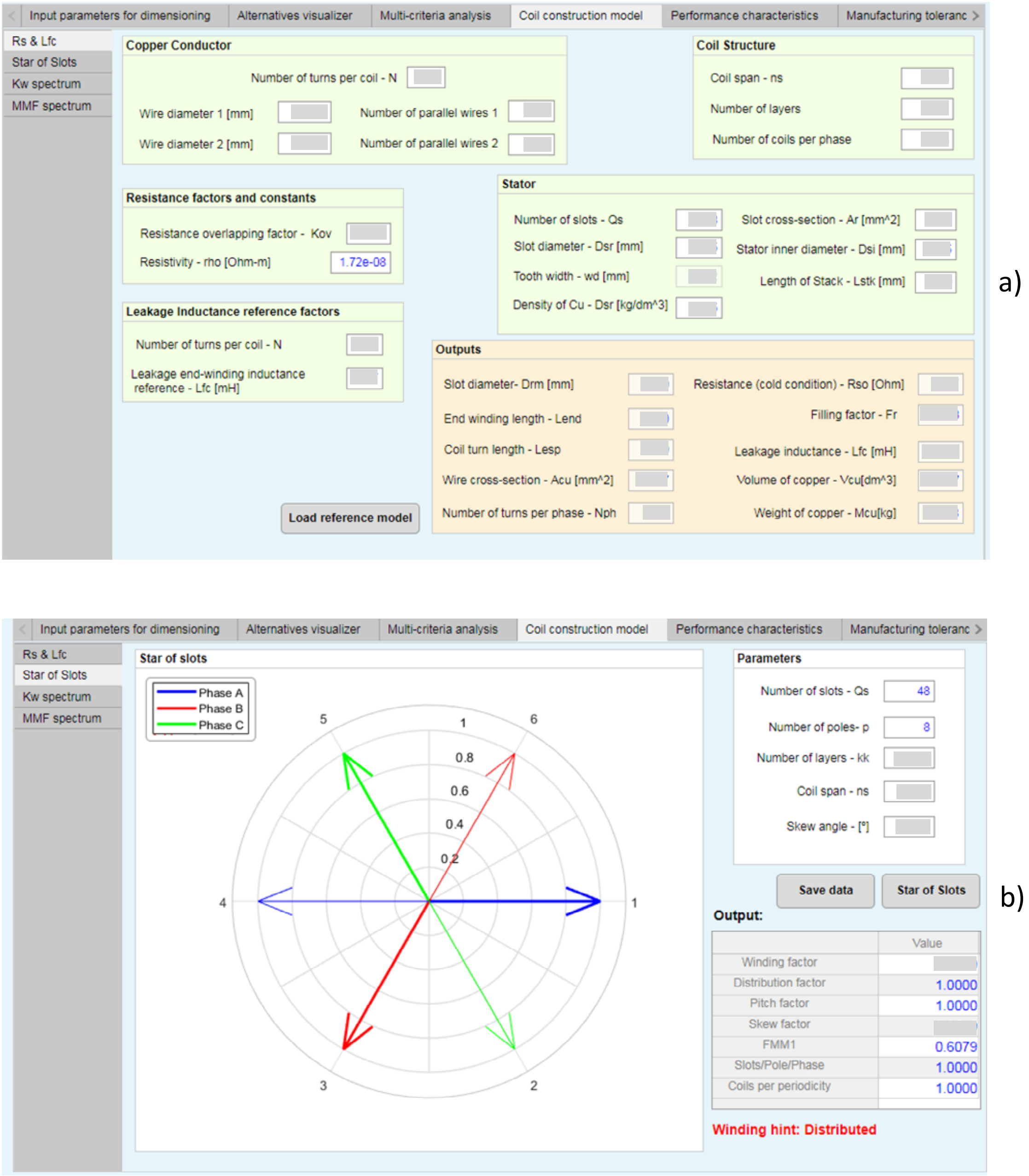


Fig. 6-25. Domain-specific design stage: Multi-criteria analysis tab, TOPSIS example.

The next step is to adjust the coil structure and check the winding model. Fig. 6-26 and Fig. 6-27 shows the four tabs available for the designer to accomplish the task. Fig. 6-26a shows the Rs & Lfc tab, here the designer must define the wire diameter and number of wires in parallel in order to get the filling factor used to obtain the motor alternatives. The process could also be backwards, first obtain a filling factor based on the define coils to get the alternative options (decision that should be made by the designer). The tab works similarly to a spreadsheet where any change to a value in a textbox affects the others and the changes are reflected automatically. Fig. 6-26b shows the Star of Slots tab where the user can check the winding parameters. The KBE application suggests the type of winding to apply based on the motor. Fig. 6-27c and Fig. 6-27d, shows the winding factor spectrum and the magnetomotive force spectrum and waveforms. The micro-level KBE application running for this case is WindingComp.



Resistivity - rho [Ohm-m] 1.72e-08

Number of slots - Qs 48

Number of poles- p 8

Output:

	Value
Winding factor	
Distribution factor	1.0000
Pitch factor	1.0000
Skew factor	
FMM1	0.6079
Slots/Pole/Phase	1.0000
Coils per periodicity	1.0000

Winding hint: Distributed

Fig. 6-26. Domain-specific design stage: Coil construction model tab; a) coil physical structure; b) star of slots and winding parameters;

After setting the coil construction model then the selected motor has to be entered in the selected motor tab and the performance characteristic loaded by clicking load. Once this is done, the table of performance characteristic and volume, weight and cost is filled out as illustrated in Fig. 6-28. The designer can adjust any value of the latter table and can also add any description.

The next step is to satisfy the torque ripple requirement, check Fig. 6-29a. the designer has to enter the harmonic number corresponding to the frequency he wants to abolish of the torque.

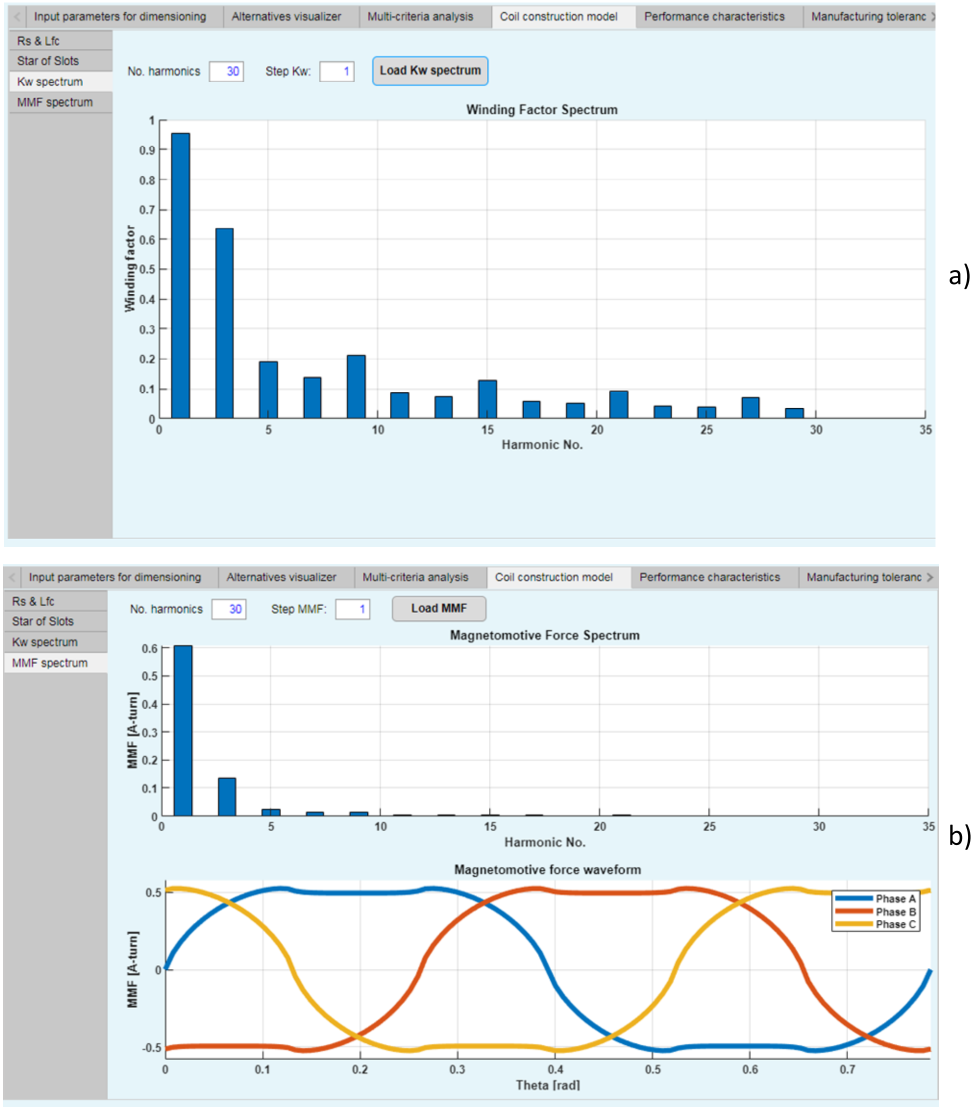
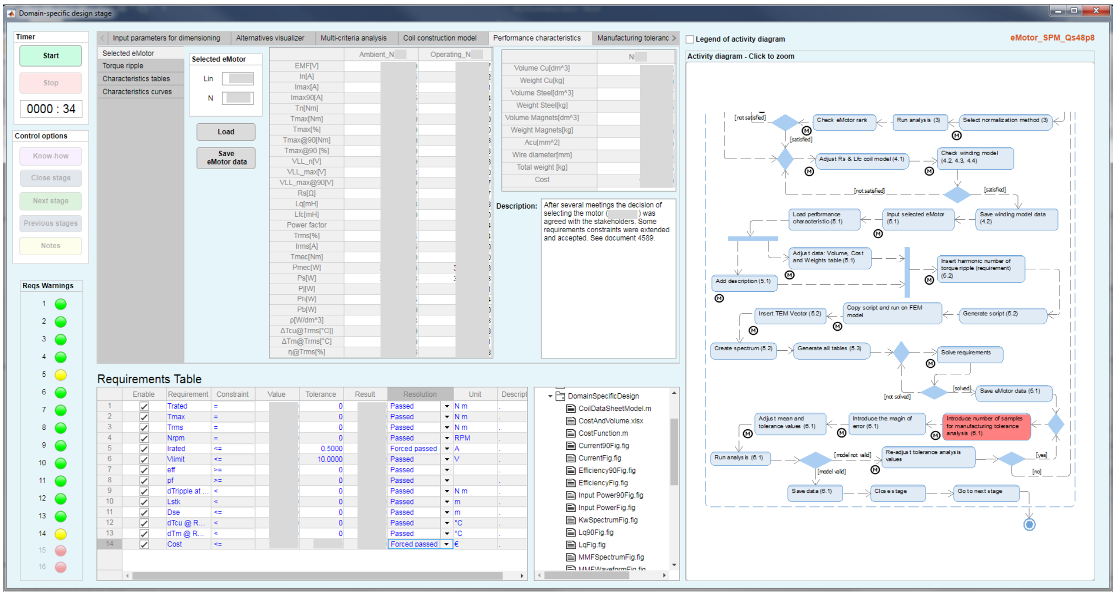

No. harmonics: 30
Step Kw: 1
No. harmonics: 30
Step MMF: 1

Fig. 6-27. Domain-specific design stage: Coil construction model tab; a) winding factor spectrum; b) MMF spectrum.

The KBE application generates automatically a script using the eMachine Operating limits algorithms described in Chapter 5. Then, the generated script is executed within Flux. The output of the script is the flux of phase A and the electromagnetic torque. These must be entered in their corresponding textbox for the spectrum generation. Flux A is only entered if the motor has skew if it does not then only the electromagnetic torque is required. Then, the torque spectrum can be created (create spectrum button) and the user can verify visually if the desired harmonic was reduced or eliminated. However, the tool automatically fills out the cell corresponding to torque ripple of the column results of the requirements table. After this, the performance tables can be generated as illustrated in Fig. 6-29b, the performance tables are: no load, load at rated speed, and load at ninety percent of rated speed. The purpose of these performance tables is to make a comparison with experimental values in the prototype-testing phase. In addition, the KBE application generates curves of the performance parameters that the designer can analyze, see Fig. 6-29c.



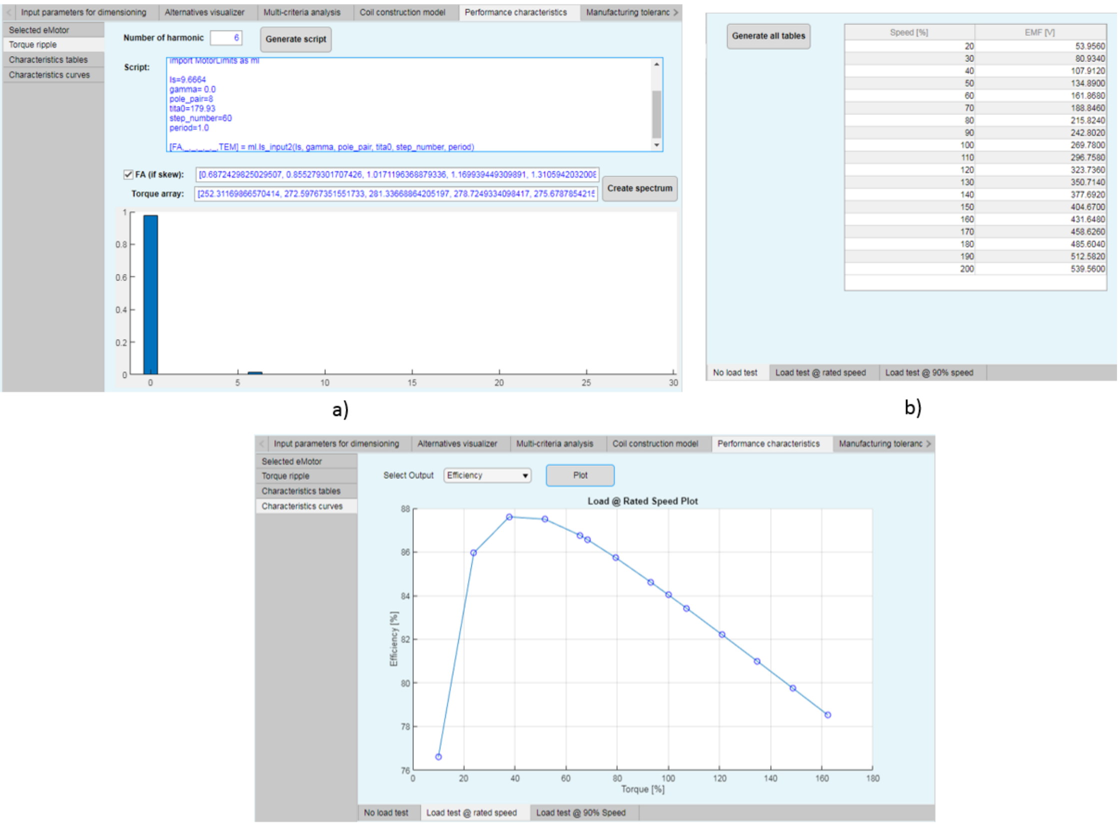
Description: After several meetings the decision of selecting the motor ([redacted]) was agreed with the stakeholders. Some requirements constraints were extended and accepted. See document 4589.

Requirements Table

	Enable	Requirement	Constraint	Value	Tolerance	Result	Resolution	Unit	Descript
1	☑	Trated	=	[redacted]	0	[redacted]	Passed	N m	.
2	☑	Tmax	=	[redacted]	0	[redacted]	Passed	N m	.
3	☑	Trms	=	[redacted]	0	[redacted]	Passed	N m	.
4	☑	Nrpm	=	[redacted]	0	[redacted]	Passed	RPM	.
5	☑	Irated	<=	[redacted]	0.5000	[redacted]	Forced passed	A	.
6	☑	Vlimit	<=	[redacted]	10.0000	[redacted]	Passed	V	.
7	☑	eff	>=	[redacted]	0	[redacted]	Passed		.
8	☑	pf	>=	[redacted]	0	[redacted]	Passed		.
9	☑	dTripple at …	<	[redacted]	0	[redacted]	Passed	N m	.
10	☑	Lstk	<	[redacted]	0	[redacted]	Passed	m	.
11	☑	Dse	<=	[redacted]	0	[redacted]	Passed	m	.
12	☑	dTcu @ R…	<	[redacted]	0	[redacted]	Passed	°C	.
13	☑	dTm @ R…	<	[redacted]	0	[redacted]	Passed	°C	.
14	☑	Cost	<=	[redacted]	[redacted]	[redacted]	Forced passed	€	.

DomainSpecificDesign
CoilDataSheetModel.m
CostAndVolume.xlsx
CostFunction.m
Current90Fig.fig
CurrentFig.fig
Efficiency90Fig.fig
EfficiencyFig.fig
Input Power90Fig.fig
Input PowerFig.fig
KwSpectrumFig.fig
Lq90Fig.fig
LqFig.fig
MMFSpectrumFig.fig

Fig. 6-28. Domain-specific design stage: Performance characteristics tab, selected eMotor.



Number of harmonic: 6

Script:

```
import MotorLimits as ml

Is=9.6664
gamma= 0.0
pole_pair=8
tita0=179.93
step_number=60
period=1.0

[FA,_,_,_,_,_,TEM] = ml.Is_input2(Is, gamma, pole_pair, tita0, step_number, period)
```

☑ FA (if skew): [0.6872429825029507, 0.855279301707426, 1.0171196368879336, 1.169939449309891, 1.3105942032008

Torque array: [252.31169866570414, 272.59767351551733, 281.33668864205197, 278.7249334098417, 275.6787854215

a)

Speed [%]	EMF [V]
20	53.9560
30	80.9340
40	107.9120
50	134.8900
60	161.8680
70	188.8460
80	215.8240
90	242.8020
100	269.7800
110	296.7580
120	323.7360
130	350.7140
140	377.6920
150	404.6700
160	431.6480
170	458.6260
180	485.6040
190	512.5820
200	539.5600

b)

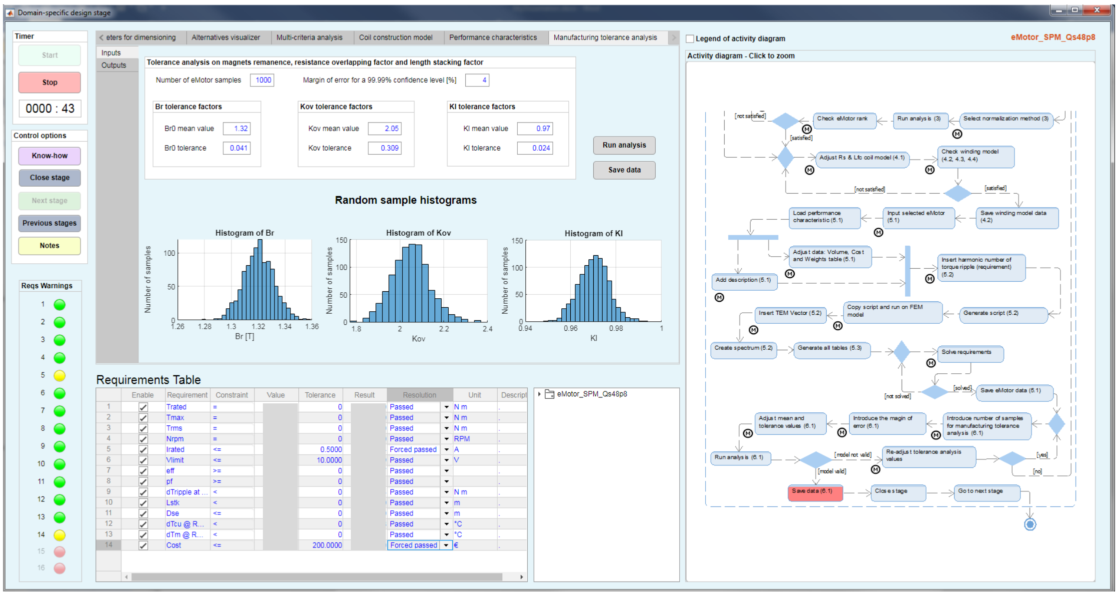
Select Output: Efficiency

c)

Fig. 6-29.Domain-specific design stage: a) torque ripple analysis tab; b) generated characteristics tables c) plotted characteristics curves

Along all the aforementioned process in the performance characteristics tab the KBE application fills automatically the requirement table with the results and selects the proper resolution (no result, passed, failed, forced passed) according to their constraint. In the example provided in Fig. 6-28, requirement 5 (Irated) and 14 (cost) are not satisfied. They are above the limit, hence, a failed resolution and a red lamp should appeared for both requirements. Nevertheless, the resolution was manually changed to "forced passed" and the requirement lamps changed to yellow to record and trace the event as forced. In the description field, the designer can add any information about this issue, as shown in the same figure. This example was intentionally prepared for the reader to appreciate this capability.

The last step of the design phase is to check the manufacturing tolerance based on three design variables, Br, Kl, and Kov. The selection of these variables is a consequence of a sensitivity analysis studied by Gomez in [91]. In addition, the tool is explained in more detail in the same reference. In summary, it analyzes statistical distributions of the machine performance considering serial-production tolerance variations, therefore obtaining critical values that supports the designer on the decision whether the design is valid or not. The micro-level KBE application used as user interface commands. For the macro-level KBE application a GUI was created and shown in Fig. 6-30.



Tolerance analysis on magnets remanence, resistance overlapping factor and length stacking factor

Number of eMotor samples: 1000
Margin of error for a 99.99% confidence level [%]: 4

Br tolerance factors
Br0 mean value: 1.32
Br0 tolerance: 0.041

Kov tolerance factors
Kov mean value: 2.05
Kov tolerance: 0.309

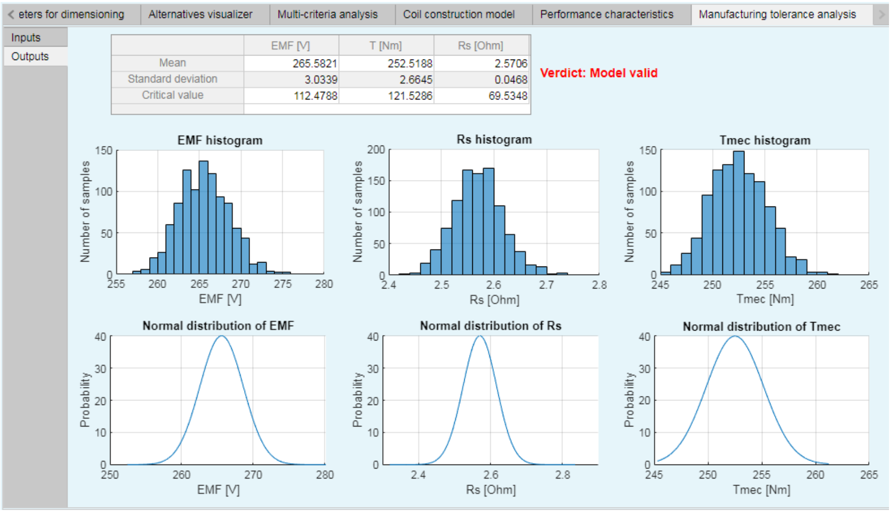
Kl tolerance factors
Kl mean value: 0.97
Kl tolerance: 0.024

Requirements Table

	Enable	Requirement	Constraint	Value	Tolerance	Result	Resolution	Unit	Descript
1	✓	Trated	=		0		Passed	N m	.
2	✓	Tmax	=		0		Passed	N m	.
3	✓	Trms	=		0		Passed	N m	.
4	✓	Nrpm	=		0		Passed	RPM	.
5	✓	Irated	<=		0.5000		Forced passed	A	.
6	✓	Vlimit	<=		10.0000		Passed	V	.
7	✓	eff	>=		0		Passed		.
8	✓	pf	>=		0		Passed		.
9	✓	dTripple at ...	<		0		Passed	N m	.
10	✓	Lstk	<		0		Passed	m	.
11	✓	Dse	<=		0		Passed	m	.
12	✓	dTcu @ R...	<		0		Passed	°C	.
13	✓	dTm @ R...	<		0		Passed	°C	.
14	✓	Cost	<=		200.0000		Forced passed	€	.

Fig. 6-30. Domain-specific design stage: Manufacturing tolerance analysis tab, inputs.

The designer must entered the number of samples units (1000 for this case), and the margin of error for a 99.99% confidence level (4% in this case). In addition, the mean values of each design parameter and the tolerance for each parameter. After this, the analysis can be executed and the data saved. The time of execution depends on the number of samples. The results of the analysis are shown in Fig. 6-31. According to [91] if the critical value is greater than 3.62 the design model can be affirmed as valid. On the other hand, if the critical value is below 3.62 the defective machines should be obtained through the standard normal distribution probabilities.

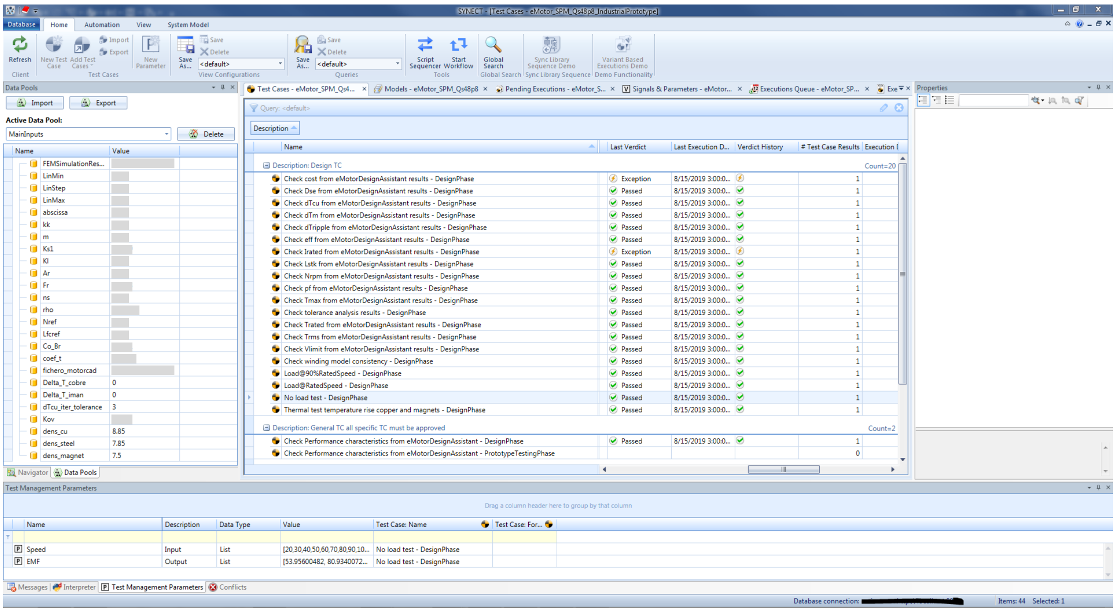


	EMF [V]	T [Nm]	Rs [Ohm]
Mean	265.5821	252.5188	2.5706
Standard deviation	3.0339	2.6645	0.0468
Critical value	112.4788	121.5286	69.5348

Verdict: Model valid

Fig. 6-31. Domain-specific design stage: manufacturing tolerance analysis tab, outputs.

Finally, the domain-specific design stage is closed and the designer can proceed to the next stage. Following the KBE system diagram of Fig. 6-3, the information is saved in the corresponding PDM system. For instance in Synect, the test cases are updated with the corresponding results to make the information available among the stakeholders. Fig. 6-32 shows a screenshot of Synect test cases updated with input and output parameters as entered and obtained respectively in the KBE application. The test cases are linked to their corresponding requirement to ensure coverage of resolution. In the same way, values and resolution for the design phase is updated in Rhapsody (Fig. 6-7) and Doors (Fig. 6-6).



Active Data Pool: MainInputs

Name	Value
FEMSimulationRes...	[redacted]
LinMin	[redacted]
LinStep	[redacted]
LinMax	[redacted]
abscissa	[redacted]
kk	[redacted]
m	[redacted]
Ks1	[redacted]
Kl	[redacted]
Ar	[redacted]
Fr	[redacted]
ns	[redacted]
rho	[redacted]
Nref	[redacted]
Lfcref	[redacted]
Co_Br	[redacted]
coef_t	[redacted]
fichero_motorcad	[redacted]
Delta_T_cobre	0
Delta_T_iman	0
dTcu_iter_tolerance	3
Kov	[redacted]
dens_cu	8.85
dens_steel	7.85
dens_magnet	7.5

Name	Last Verdict	Last Execution D...	Verdict History	# Test Case Results
Description: Design TC				Count=20
Check cost from eMotorDesignAssistant results - DesignPhase	Exception	8/15/2019 3:00:0...	⚡	1
Check Dse from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check dTcu from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check dTm from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check dTripple from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check eff from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check Irated from eMotorDesignAssistant results - DesignPhase	Exception	8/15/2019 3:00:0...	⚡	1
Check Lstk from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check Nrpm from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check pf from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check Tmax from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check tolerance analysis results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check Trated from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check Trms from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check Vlimit from eMotorDesignAssistant results - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check winding model consistency - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Load@90%RatedSpeed - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Load@RatedSpeed - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
No load test - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Thermal test temperature rise copper and magnets - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Description: General TC all specific TC must be approved				Count=2
Check Performance characteristics from eMotorDesignAssistant - DesignPhase	Passed	8/15/2019 3:00:0...	✓	1
Check Performance characteristics from eMotorDesignAssistant - PrototypeTestingPhase				0

Name	Description	Data Type	Value	Test Case: Name	Test Case: For...
Speed	Input	List	[20,30,40,50,60,70,80,90,10...	No load test - DesignPhase	
EMF	Output	List	[53.95600482, 80.9340072...	No load test - DesignPhase	

Database connection: [redacted]
Items: 44 Selected: 1

Fig. 6-32. Synect test case management linked to requirements

The animated guideline for the domain-specific design stage is shown in Fig. 6-33.

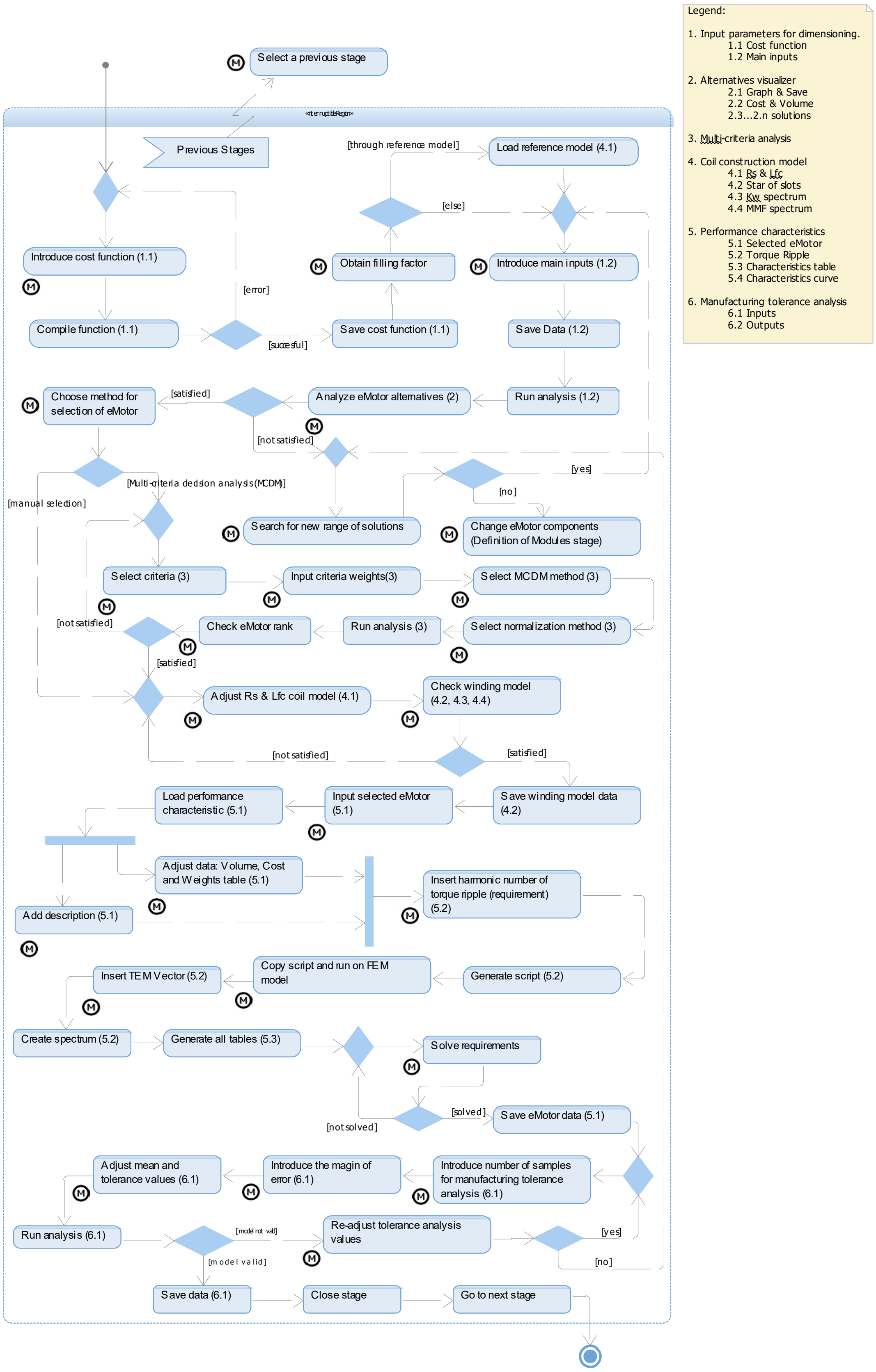
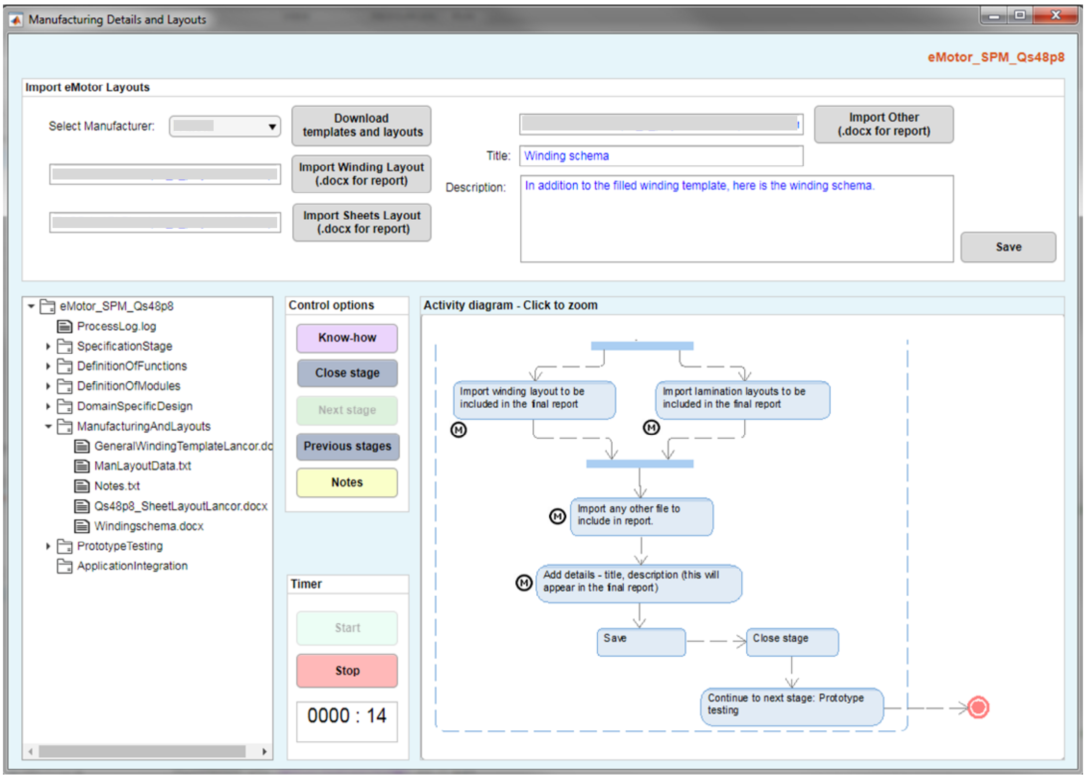


Fig. 6-33. Guideline embedded in the KBE application for the domain-specific design stage.

6.3.5 Manufacturing details and layouts stage

In this stage, the designer selects one manufacturer from the list to download the templates and layouts available from that particular manufacturer. After filling the templates and customizing the layouts properly, he can import them. Importing the templates and layouts through the KBE application help to keep record of the process and the imported files become part of the final design report. All files are added to the stage folder and can be visualized in the explorer. Fig. 6-34 presents a screenshot of the user interface.

Fig. 6-35 shows the animated guideline for this phase with the step that the user should follow. Finally, the stage can be closed and the designer can proceed to the next stage. The information of the generated templates and layouts are also saved in Aras for the specific project. In this work, the templates and layouts are not shown to the reader due to confidentiality reasons.



Title: Winding schema

Description: In addition to the filled winding template, here is the winding schema.

Fig. 6-34.User interface of manufacturing details and layouts stage.

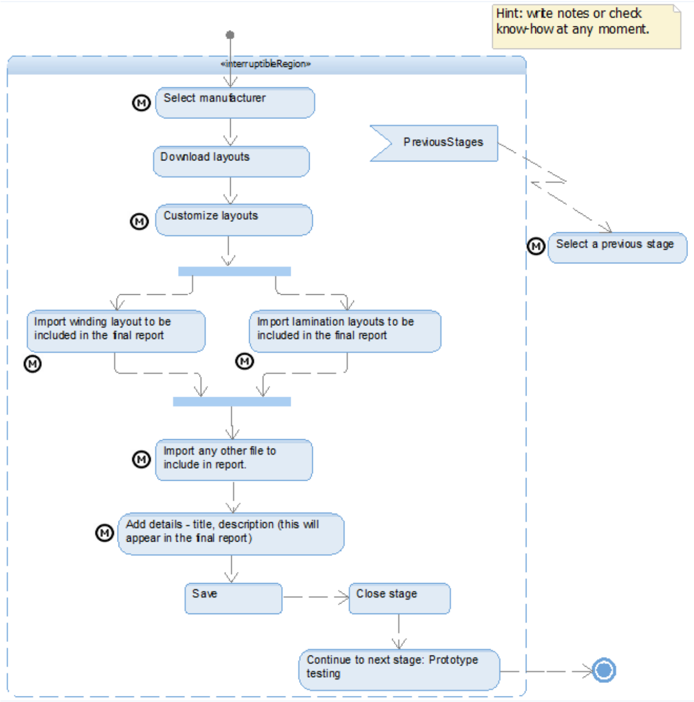


Fig. 6-35. Guideline of manufacturing details and layouts stage.

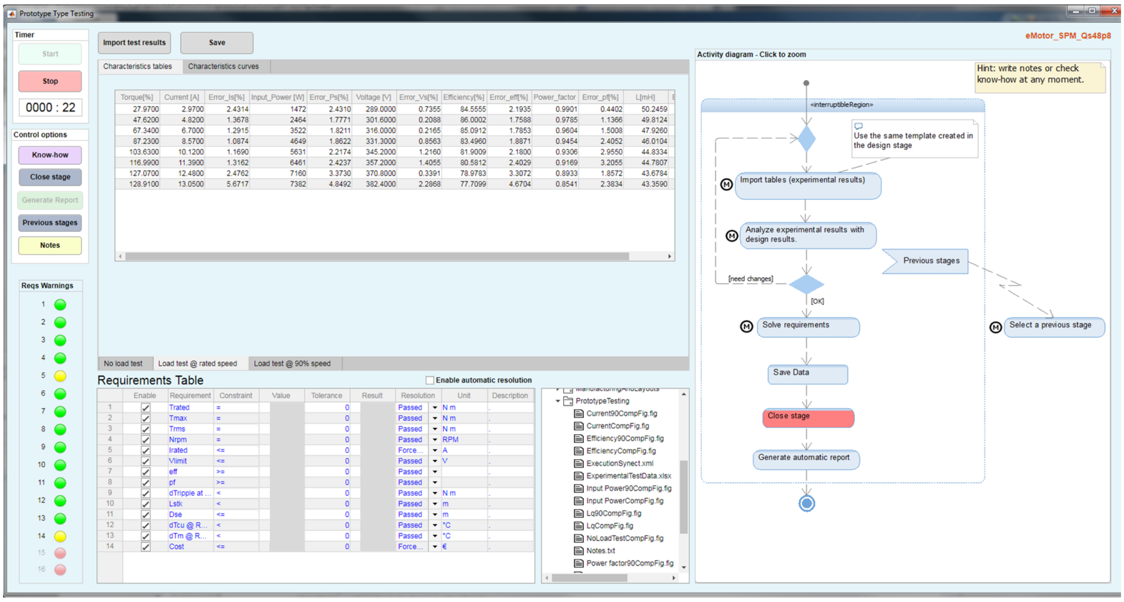
6.3.6 Prototype testing stage

This is the last stage where the macro-level KBE application brings support to the development process. In this stage, the experimental tests are carried out in workbench and the information saved in Synect test cases with the input parameters for assurance of future replications if required.

Before obtaining the experimental results, test acceptance criteria for validation of each test should be set as explained. For instance, a checklist with fields assuring the proper mounting of the motor on the workbench, noise and vibration levels under limits and others. Furthermore, definition of acceptance criteria of experimental results for each test should be set in order to confirm the data obtained is coherent with the expected results. The aforementioned aspects are not supported by the KBE application but it is interesting to consider them in future work.

After the experimental tests are carried out the test data can be imported using the KBE application, the same format created in the design phase for the selected motor results has to be used. Only the no load test, test at rated speed and test at ninety percent rated speed are processed by the KBE application for comparison analysis. The designer has to enter in the requirement table the values for the torque ripple and temperature rise requirements obtained from their respective tests, the rest of requirements are filled out automatically. Fig. 6-36 shows the user interface of the prototype testing phase, the figure shows the experimental data with the error between the experimental data and design data computed by the KBE application. In addition, the

KBE application also provides the capability of viewing the comparison between design results in FEM and tests results through plotted curves for each characteristic (current, voltage, power, efficiency, etc.). One example is illustrated in Fig. 6-37.



Torque[%]	Current [A]	Error_Is[%]	Input_Power [W]	Error_Ps[%]	Voltage [V]	Error_Vs[%]	Efficiency[%]	Error_eff[%]	Power_factor	Error_pf[%]	L[mH]
27.9700	2.9700	2.4314	1472	2.4310	289.0000	0.7355	84.5555	2.1935	0.9901	0.4402	50.2459
47.6200	4.8200	1.3678	2464	1.7771	301.6000	0.2088	86.0002	1.7588	0.9785	1.1366	49.8124
67.3400	6.7000	1.2915	3522	1.8211	316.0000	0.2165	85.0912	1.7853	0.9604	1.5008	47.9260
87.2300	8.5700	1.0874	4649	1.8622	331.3000	0.8563	83.4960	1.8871	0.9454	2.4052	46.0104
103.6300	10.1200	1.1690	5631	2.2174	345.2000	1.2160	81.9009	2.1800	0.9306	2.9550	44.8334
116.9900	11.3900	1.3162	6461	2.4237	357.2000	1.4055	80.5812	2.4029	0.9169	3.2055	44.7807
127.0700	12.4800	2.4762	7160	3.3730	370.8000	0.3391	78.9783	3.3072	0.8933	1.8572	43.6784
128.9100	13.0500	5.6717	7382	4.8492	382.4000	2.2868	77.7099	4.6704	0.8541	2.3834	43.3590

Requirements Table

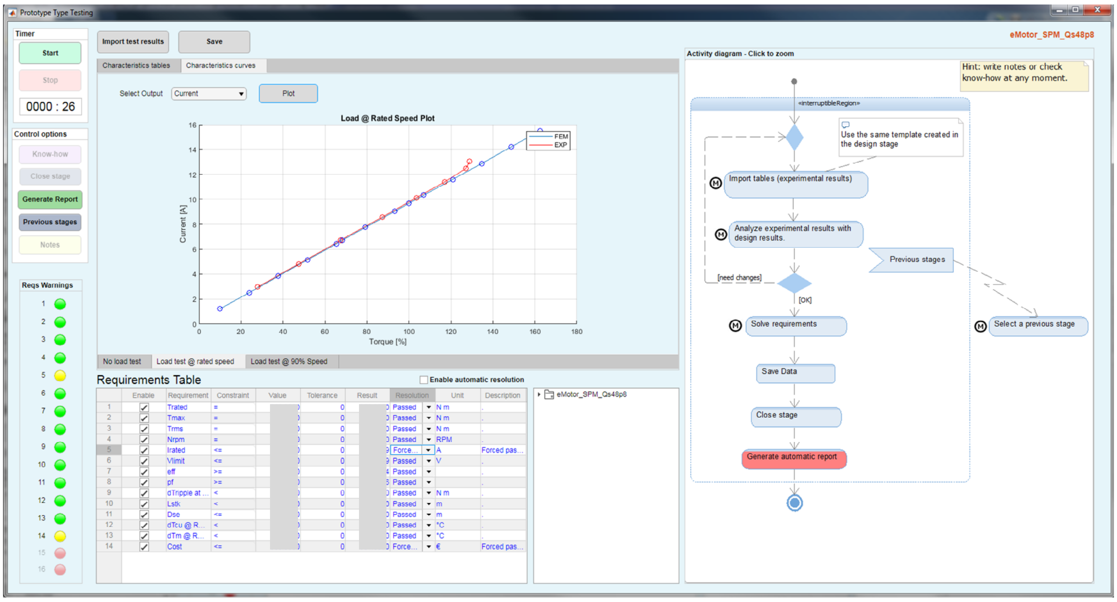
☐ Enable automatic resolution

	Enable	Requirement	Constraint	Value	Tolerance	Result	Resolution	Unit	Description
1	☑	Trated	=	[redacted]	0	[redacted]	Passed	N m	.
2	☑	Tmax	=	[redacted]	0	[redacted]	Passed	N m	.
3	☑	Trms	=	[redacted]	0	[redacted]	Passed	N m	.
4	☑	Nrpm	=	[redacted]	0	[redacted]	Passed	RPM	.
5	☑	Irated	<=	[redacted]	0	[redacted]	Force...	A	.
6	☑	Vlimit	<=	[redacted]	0	[redacted]	Passed	V	.
7	☑	eff	>=	[redacted]	0	[redacted]	Passed		.
8	☑	pf	>=	[redacted]	0	[redacted]	Passed		.
9	☑	dTripple at ...	<	[redacted]	0	[redacted]	Passed	N m	.
10	☑	Lstk	<	[redacted]	0	[redacted]	Passed	m	.
11	☑	Dse	<=	[redacted]	0	[redacted]	Passed	m	.
12	☑	dTcu @ R...	<	[redacted]	0	[redacted]	Passed	°C	.
13	☑	dTm @ R...	<	[redacted]	0	[redacted]	Passed	°C	.
14	☑	Cost	<=	[redacted]	0	[redacted]	Force...	€	.

Fig. 6-36. User interface of prototype testing stage, characteristics tables tab, load test @ rated speed.

Finally, the user fills out the requirements results field and resolution in order to verify the requirements in this phase.

After closing the stage, the generate report button is enabled and the KBE application ends its support by creating a detailed report for the designer with the process he followed in all the development process.



Requirements Table

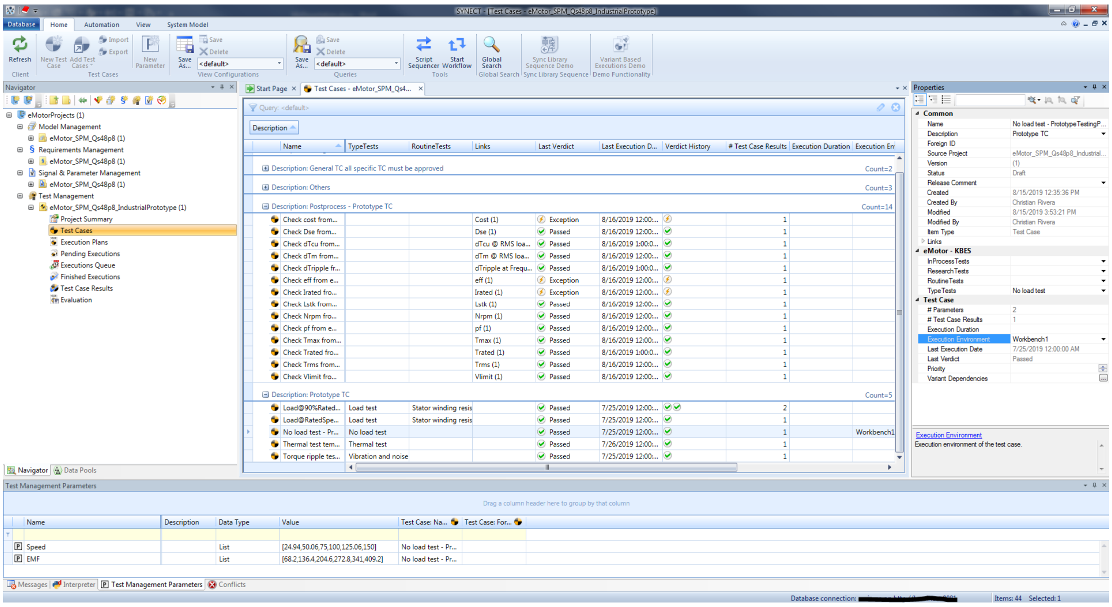
☐ Enable automatic resolution

	Enable	Requirement	Constraint	Value	Tolerance	Result	Resolution	Unit	Description
1	☑	Trated	=	[redacted]	0	[redacted]	Passed	N m	.
2	☑	Tmax	=	[redacted]	0	[redacted]	Passed	N m	.
3	☑	Trms	=	[redacted]	0	[redacted]	Passed	N m	.
4	☑	Nrpm	=	[redacted]	0	[redacted]	Passed	RPM	.
5	☑	Irated	<=	[redacted]	0	[redacted]	Force...	A	Forced pas...
6	☑	Vlimit	<=	[redacted]	0	[redacted]	Passed	V	.
7	☑	eff	>=	[redacted]	0	[redacted]	Passed		.
8	☑	pf	>=	[redacted]	0	[redacted]	Passed		.
9	☑	dTripple at ...	<	[redacted]	0	[redacted]	Passed	N m	.
10	☑	Lstk	<	[redacted]	0	[redacted]	Passed	m	.
11	☑	Dse	<=	[redacted]	0	[redacted]	Passed	m	.
12	☑	dTcu @ R...	<	[redacted]	0	[redacted]	Passed	°C	.
13	☑	dTm @ R...	<	[redacted]	0	[redacted]	Passed	°C	.
14	☑	Cost	<=	[redacted]	0	[redacted]	Force...	€	Forced pas...

Fig. 6-37.User interface of prototype testing stage, characteristics curves tab, load test @ rated speed.

Similar to the design phase, the last step is to resolve the test cases in Synect, update the test values in Rhapsody, and update Doors resolution. Fig. 6-38 shows the test cases classification in Synect with their respective resolution.

The KBE system leverage all the integrated technologies to keep all the information at hand making the entire process reliable and replicable.



Name	TypeTests	RoutineTests	Links	Last Verdict	Last Execution D...	Verdict History	# Test Case Results	Execution Duration	Execution En
Description: General TC all specific TC must be approved									Count=2
Description: Others									Count=3
Description: Postprocess - Prototype TC									Count=14
Check cost from...			Cost (1)	Exception	8/16/2019 12:00:...	⚡	1		
Check Dse from...			Dse (1)	Passed	8/16/2019 12:00:...	✓	1		
Check dTcu from...			dTcu @ RMS loa...	Passed	8/16/2019 1:00:0...	✓	1		
Check dTm from...			dTm @ RMS loa...	Passed	8/16/2019 12:00:...	✓	1		
Check dTripple fr...			dTripple at Frequ...	Passed	8/16/2019 1:00:0...	✓	1		
Check eff from e...			eff (1)	Exception	8/16/2019 12:00:...	⚡	1		
Check Irated fro...			Irated (1)	Exception	8/16/2019 12:00:...	⚡	1		
Check Lstk from...			Lstk (1)	Passed	8/16/2019 12:00:...	✓	1		
Check Nrpm fro...			Nrpm (1)	Passed	8/16/2019 12:00:...	✓	1		
Check pf from e...			pf (1)	Passed	8/16/2019 12:00:...	✓	1		
Check Tmax from...			Tmax (1)	Passed	8/16/2019 12:00:...	✓	1		
Check Trated fro...			Trated (1)	Passed	8/16/2019 1:00:0...	✓	1		
Check Trms from...			Trms (1)	Passed	8/16/2019 12:00:...	✓	1		
Check Vlimit fro...			Vlimit (1)	Passed	8/16/2019 12:00:...	✓	1		
Description: Prototype TC									Count=5
Load@90%Rated...	Load test	Stator winding resis		Passed	7/25/2019 12:00:...	✓✓	2		
Load@RatedSpe...	Load test	Stator winding resis		Passed	7/25/2019 12:00:...	✓	1		
No load test - Pr...	No load test			Passed	7/25/2019 12:00:...	✓	1		Workbench1
Thermal test tem...	Thermal test			Passed	7/26/2019 12:00:...	✓	1		
Torque ripple tes...	Vibration and noise			Passed	7/25/2019 12:00:...	✓	1		

Properties	
Common	
Name	No load test - PrototypeTestingP...
Description	Prototype TC
Foreign ID	
Source Project	eMotor_SPM_Qs48p8_Industrial...
Version	(1)
Status	Draft
Release Comment	
Created	8/15/2019 12:35:36 PM
Created By	Christian Rivera
Modified	8/15/2019 3:53:21 PM
Modified By	Christian Rivera
Item Type	Test Case
Links	
eMotor - KBES	
InProcessTests	
ResearchTests	
RoutineTests	
TypeTests	No load test
Test Case	
# Parameters	2
# Test Case Results	1
Execution Duration	
Execution Environment	Workbench1
Last Execution Date	7/25/2019 12:00:00 AM
Last Verdict	Passed
Priority	
Variant Dependencies	

Execution Environment
Execution environment of the test case.

Test Management Parameters

Drag a column header here to group by that column

Name	Description	Data Type	Value	Test Case: Na...	Test Case: For...
Speed		List	[24.94,50.06,75,100,125.06,150]	No load test - Pr...	
EMF		List	[68.2,136.4,204.6,272.8,341,409.2]	No load test - Pr...	

Database connection: [redacted] Items: 44 Selected: 1

Fig. 6-38. Synect test cases classification and resolution in prototype testing stage.

Fig. 6-35 shows the animated guideline for this phase with the step that the user should follow.

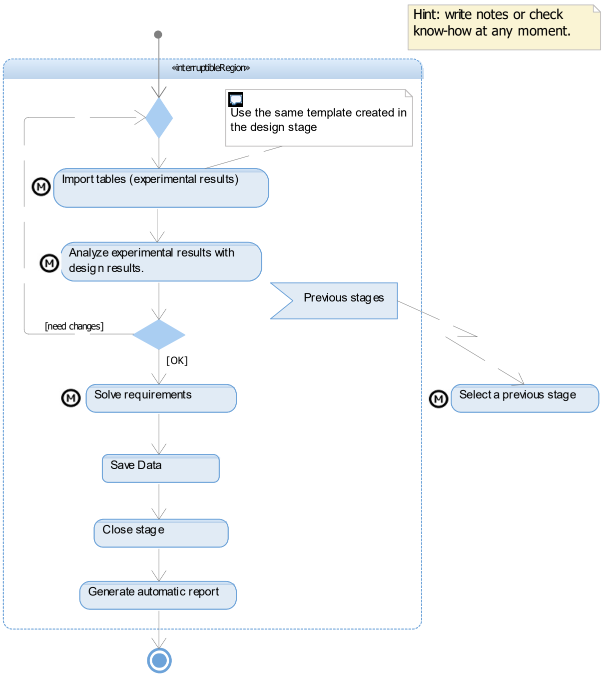


Fig. 6-39. Guideline of prototype testing stage.

6.3.7 Statistics of the eMotor development process

The outcome of the process is a viable motor solution for the given requirements. To elaborate more about the final design report, this is generated for the designer showing the data he used in each step including drawings, forms, curves, requirement, alternative solutions, notes. The handled information allows replicating any phase. In other words, no information is lost in the process.

In addition, the final report also includes a summary of metrics such as time spent on each stage, start and ending dates, revisits to each stage, and the total time of the project from its opening until its closure. Allowing measuring the performance of the designer with the KBE system through this data. All these data is retrieved automatically from the log file the application creates to record all the events throughout the development process. For the process followed throughout this chapter, these metrics are presented in Table 6-1.

Table 6-1. Metrics of the development process.

Description	Value
Project starting date	12-Aug-2019 17:17:25
Project ending date	13-Aug-2019 14:58:08
Working folder	-Omitted-
Specification stage starting date	12-Aug-2019 17:18:11
Specification stage ending date	12-Aug-2019 17:27:33
Time elapsed on specification stage	0000 : 09
Revisits to specification stage	0
Def. functions & structure stage starting date	12-Aug-2019 17:28:40
Def. functions & structure stage ending date	12-Aug-2019 17:31:41
Time elapsed on def. of functions & structures stage	0000 : 03
Revisits to def. of functions & structure stage	0
Definition of modules stage starting date	13-Aug-2019 09:45:50
Definition of modules stage ending date	13-Aug-2019 09:48:04
Time elapsed on definition of modules stage	0000 : 02
Revisits to definition of modules stage	0
Design stage starting date	13-Aug-2019 12:15:04
Design stage ending date	13-Aug-2019 13:00:07
Time elapsed on design stage	0000 : 45
Revisits to design stage	0
Man. details and layouts starting date	13-Aug-2019 14:13:47
Man. details and layouts ending date	13-Aug-2019 14:28:21
Time elapsed on man. details and layouts stage	0000 : 14
Revisits to man. details and layouts	0
Prototype type testing starting date	13-Aug-2019 14:30:25
Prototype type testing ending date	13-Aug-2019 14:56:38
Time elapsed on Prototype type testing stage	0000 : 26
Total time	1 hr. 39 min.

A comparison of working time consumption with and without the KBE system is made for a straightforward design without time spent due to changes in requirements or any other tasks that could slow the process. One expert was interviewed on how much time he spends on each stage without considering any problems i.e. an ideal process. The metrics showed in Table 6-1 are used to compare the working time spent by the interviewed expert without the KBE system.

Table 6-2 shows the time consumption comparison. The time consumption given by the expert are estimations; however, they reflect the advantage of implementing the KBE system by obtaining a reduction of up to 76%.

The major reasons for a faster development process is not only to automation of repetitive tasks but also that the information is at hand in every stage i.e. the designer has not to spend time searching for any type of data and information. It is important to remark the time showed in the table is working time spent on the steps presented in the activity diagrams (guidelines) on each stage, so tasks outside these actions are not considered. For instance, the prototype testing stage involves also making the tests, which requires many hours of work especially for thermal tests. Therefore, the time presented in this stage is only time spent on the analysis of data as presented in the activity diagram in the prototype testing stage. The same applies for the rest of stages.

Table 6-2. Working time consumption comparison between expert using and not using the KBE system.

Stages	Without KBE system	With KBE system
Specification stage	30 min.	9 min.
Definition of functions structure stage	30 min.	3 min.
Definition of modules	30 min.	2 min.
Design stage	240 min.	45 min.
Manufacturing details and layouts	30 min.	14 min.
Prototype testing	60 min.	26 min.
Application integration testing	-	-
Total time	7 hr.:00 min.	1 hr. 39 min.

6.4 Discussion

A complete description of the developed KBE system has been provided. This KBE system ensures all knowledge, information and data is not lost through the whole development process. In summary, the KBE system provides the functionalities of requirement management, development process traceability, knowledge accessibility, automation of tasks, and intelligent support.

Although not shown, one additional feature to highlight is that the designer has the possibility of creating several projects at the same time. This allows the execution of projects in parallel to compare different motor solutions based on distinct models i.e. different stator and rotor laminations.

However, there is still margin of improvement for future line of work. For instance, the KBE application does not yet automatically integrates the PDM systems. The manufacturing templates and layouts are not filled out automatically. Typically, the knowledge of type and routine tests are well known within a company, therefore, criteria for test validation, acceptance of test results and acceptance of the prototype can be annexed as functionalities in the macro-level KBE application. This will enrich the prototype testing phase since until now the current KBE system only provides the advantage of saving the test parameters by using Synect. However, this ensures test replicability.

In addition, for future line of work a study should be carried out on how to make the developed KBE system pass from the current industrial V-cycle to the component V-cycle and later to the concept V-cycle. Therefore, allowing more degrees of freedom for the design of a motor.

To prove the advantage of implementing the KBE application a time consumption comparison was made for a designer using and not using the KBE system. The implementation of the KBE application for the same tasks reduced the time workload up to 76%.

For the scope of this thesis, the successful development of a KBE system for the elevator industry demonstrated that the proposed KBE system framework actually works providing the user enough guidance for the development of a successful KBE system.

Chapter 7

CONCLUSIONS AND FUTURE WORK

This chapter reports the conclusions and highlights future work. The conclusions are made from general to specific regarding the objectives set in the introduction.

7.1 Conclusions

The research proposition of this thesis is that *a knowledge-based engineering system that manage and automate tasks of the electric machine development process will lead to development-time reduction without risking reliability or any other value driver.*

From which the main objective of this thesis was to define a knowledge-based engineering (KBE) system framework for the development of electric machines. As a general conclusion, it can be stated that a complete proposal of a KBE system framework has been developed including the following aspects:

1. A standardized macro-level framework coping with the development phases at different levels of technological maturity.
2. An adapted and extended framework for capturing and formalizing the knowledge in the domain of electric machines development process.
3. A generic knowledge base containing knowledge regarding the product model and process model that can be reused for future developments of electric machines.
4. A SW architecture proposal for fast implementation integrating commercial SW, advanced design tools, and a customized and flexible KBE application for the real needs of an electric machine manufacturer.
5. Micro and Macro level KBE applications concepts following the MVP approach.

The framework allows developing KBE systems for the electric machine industry providing capabilities such as development process traceability, knowledge management, automation of repetitive tasks and intelligent support. Substantiation of the process for each stage was presented with enough detail to allow replicability of every process. The KBE system framework is encapsulated in the so-called KBV2-model. The model consists of a knowledge base generation part and the standardized macro-level framework for the development of electric machines.

Use case examples with comparison of time consumption by using and not using the KBE application were presented. From the results obtained it can be concluded that KBE systems can reduce the development time significantly in addition to the reduction of workload to the user.

Conclusions on each derived objective are now presented.

7.1.1 Standardized macro-level framework for the development of eMachines

Inside the company, new development of electric machines start from a low TRL level (proof of concept or laboratory validation) and finishes at higher TRL level (competitive manufacturing). Taking into account the full process to obtain the technological maturity, in this thesis the process has been divided in three standardized macro-level V-cycles. For each V-cycle a specific goal is defined.

- In the **Concept V-cycle** the goal is to obtain a Virtual Concept able to be modified or adapted to different applications needs.
- In the **Component V-cycle** the goal is to obtain the basic components and manufacturing tools in order to use them for the final product design in the Industrial V-cycle.
- In the **Industrial V-cycle** the goal is to develop the final product that fulfils the specific requirements of the application of the client using standardized components with a predefined manufacturing process.

The maturity concept of the V-cycles was derived from the VDI-2206 guideline. Building several prototypes is a typical approach to increase the product maturity in an electric machine development process. Typically, the design presented in scientific literature corresponds to the Concept V-cycle and the real industrialized designs corresponds to the Industrial V-cycle. It is considered that the major contribution of this thesis regarding this issue is the proposal of a common standardized macro-level framework allowing the integration of development tasks accounting innovation, industrialization and final production. This common framework allow the direct conversion of virtual concepts into product designs maintaining traceability among the original concept, the industrialized component and the final product. The objective is that a generic virtual concept would be used in several component developments for different applications purposes. Then, the generic innovation would be easily expanded along the range of applications of the company.

7.1.2 Adapted methodology for capturing and formalizing the knowledge

In this thesis, an adaptation and extension of MOKA methodology was derived for the electric machine design process. The purpose of the derived methodology is to facilitate its implementation in the electric machine industry.

Due to the problems encountered to represent knowledge using the process model such as knowledge not available in the same KB formalization tool and the lack of knowledge representation for the macro level view. A new proposal of knowledge representation was developed for the eMachine process model. For the macro-process level, SysML was adopted and for the micro-process level, a new metamodel with stereotypes extended from SysML classes was provided and described through several examples. The major contribution regarding this issue is the micro-process metamodel from which a new profile called KBeMotorProfile was created and described with the stereotypes it contains. Examples of knowledge representation were given for each case.

In conclusion, substantiation was provided for the Knowledge Base Generation process upon the development of KBE systems for the electric machine industry.

7.1.3 A Generic Knowledge Base regarding product and process for the development of electric machines

Using the new adapted and extended MOKA methodology, an initial new Knowledge Base regarding product and process for the development of electric machine have been created. This generic Knowledge Base can be used as a starting point to create a specific Knowledge Base for the company. It is true that each company has its own knowledge but there is a general knowledge of the state of art regarding their technology that companies can directly use. The generic Knowledge Base derived from this thesis is an example of how the general knowledge of the state of art can enrich the specific knowledge of a company.

Product models were created from general knowledge of electric machines and put into practice through the metadata customization of a PDM system. On the other hand, process models were created for both the macro and micro levels. For the macro-level, a clear example is the developed requirement-engineering framework. This framework represent high-level knowledge regarding the specification stage of the development of electric machines and it is part of the generic KB. For the micro-process model, the developed intelligent algorithms to compute eMachine-operating limits as well as the algorithm to find an optimal operating point and compute the performance characteristics for brushless AC motors are part of the generic KB.

7.1.4 SW architecture proposal for KBE systems

In this thesis, an original SW architecture is proposed to develop KBE systems. The proposed architecture allows a cost-effective and flexible implementation. This approach tries to avoid the development of complex KBE systems that can require high implementation time as well as high maintenance costs. Therefore, it consists of the integration of the KBE applications with the commercial off the shelf SW (COTS) technologies used in the product development process (MBSE, requirement management, PDM, Data repository, and advanced design tools). Since the integration maintains the current tools used in the company this helps not to change so radically the process the eMachine experts carry out before the KBE system implementation. At the same time, considering that commercial SW make continuous improvement. Adapting the

KBE application to the commercial SW will improve the performance of the KBE system without requiring high SW developing efforts.

The center of the KBE system is the suite of KBE applications, these are in charge of managing the processes and data (from product and process databases, files, etc.), manage the integrated COTS as well as provide guidance to the end-user through the devised flow-chart process. The SW architecture proposal can be seen in Fig. 4-14.

7.1.5 MVP micro-level KBE applications and MVP KBE system

The proposed framework was implemented by the developing of micro-level KBE applications and a macro-level KBE system. The objective of these implementations was to obtain a Minimum Viable Product that allow detecting implementation problems and validating the proposed framework in different developing contexts. The developed MVP implementations examples demonstrated that a knowledge base approach allows implementing the SW without ambiguities since all the processes, structure, rules, and functions are well defined.

Next, the most relevant results for each MVP implementation are presented:

Micro-level implementations

KBE application "eMachine operating limits". This application use intelligent algorithms leveraging FEM high fidelity non-linear electromagnetic computations to obtain typical brushless AC motor characteristics during the design process as an alternative to analytical models or repetitive post-processing tasks after simulations. Operation points such as MTPA, FW, MTPV, and other points that cannot be targeted directly in a FEM tool can be obtained within acceptable range of precision (under 3%) and times of execution for practical purposes. It was found that at least a reduction of 30% of time can be obtained. Concluding the KBE application successfully accomplished the goal of supporting the designer to compute complex points reducing computational time and workload.

KBE application "eMotor Performance Characteristics Finder". This application demonstrated the reuse of knowledge by integrating the previous application and expanding its functionalities to convert it into a new application. This KBE application is capable of computing the performance characteristics of an eMotor (torque, power, losses, efficiency, etc.) at ambient temperature and operating temperature by integrating Flux and Motorcad CAE SW. The value-added in this application is that the computation can be done directly in the FEM tool (no post-process required by the designer) and the requested point (given in Torque@Speed@Vlimit) is found in the optimal region (MTPA, FW, MTPV) if the motor is capable of giving it otherwise it returns the maximum obtainable point. The knowledge that the KBE application use was represented by using the micro-process metamodel. This application opens new design opportunities allowing checking different design solutions (even in parallel) automatically that could be used within global iterative design processes.

Macro-level implementations

A KBE system for the industrial V-cycle for the elevator industry. The created KBE system for the industrial V-cycle provides to the designer process traceability, knowledge accessibility, automation of tasks and intelligent support. The KBE system tracks and trace the development process from start to end. The macro-level KBE application (so-called **eMotor design assistant**) integrates the knowledge saved in models with easy access to the user for context understanding. It automates complex tasks by implementing intelligent algorithms with the knowledge gathered from the experts.

Comparing the performance of this KBE system with respect to the original process a relevant time reduction was obtain. At the same time, thank you to the process traceability and that no information is lost during the development process the origin of any design problem can be identified, allowing returnin to the design task where the error occurred retaking again the design from this point. In addition to the advantage that every step can be replicated.

7.2 Future work

This thesis presents a KBE system framework for the development of electric machines. This framework has mainly conceptual and methodological content. Therefore, the next step will be directed to the specific implementation of KBE systems adapted to real needs of the electric machine industry. For these implementations, it is suggested to follow the MVP approach i.e. beginning with simple initial implementations to obtain fast feedback from designers. This feedback is considered important to fulfill the real needs of designers.

Although the results of this thesis can be used directly for real implementations. In order to improve the KBE system framework, the following research issues have been identified as a future work.

- Add **cooperative and communication guidelines among stakeholders along all the design process**. The participation of the stakeholders is partially considered in the standardization processes. However, this is an open research issue since new effective mechanisms are needed allowing the inclusion of the stakeholders contributions without adding obstacles to design tasks.
- Include for the KBE system framework managerial guidelines to **analyze costs and risks of a KBE solution** in an enterprise. Include guidelines for communication channels among actors.
- Include a software development methodology containing automatic communication and **interpretation of information between the KBE application and COTS** taking into account future outputs and functionalities of COTS.
- Develop more features in the **KBE application** for the **prototype testing and manufacturing phases**.
- **Develop KBE systems for the component V-cycle and the concept V-cycle**. In these V-cycles, the designer has more degrees of freedom. Therefore, micro-level processes have to offer a wider range of design space.
- Develop **new advanced features for intelligent design assistance**. Technologies using Big Data (data mining, machine learning, etc.) allows the implementation of artificial intelligence in a greater scale for searching and learning by finding patterns from huge historic datasets from the company. For instance, this could be implemented for intelligent search of solutions (for example using clustering to recover the best similar design solution or problem to the current design task),

intelligent support functions (learning from previous solutions and advising that certain parameters could have a good or bad success rate).

REFERENCES

[1] J. M. Morgan and J. K. Liker, *The Toyota Product Development System*. Productivity Press, 2008.

[2] "Model-Based Systems Engineering (MBSE) (glossary) - SEBoK," 2016. [Online]. Available: http://sebokwiki.org/wiki/Model-Based_Systems_Engineering_(MBSE)_(glossary). [Accessed: 02-Nov-2016].

[3] E. J. Reddy, C. N. V. Sridhar, and V. P. Rangadu, "Knowledge Based Engineering: Notion, Approaches and Future Trends," *Am. J. Intell. Syst.*, vol. 5, no. 1, pp. 1–17, 2015.

[4] A. Ćatic and M. Johan, "TOWARDS INTEGRATION OF KBE AND PLM," vol. 1, no. August, pp. 1–12, 2007.

[5] W. L. Chen, S. Q. Xie, and F. F. Zeng, "An integrated product development platform based on STEP and PDM," in *2009 IEEE International Conference on Industrial Engineering and Engineering Management*, 2009, pp. 1195–1199.

[6] W. Skarka, "Application of MOKA methodology in generative model creation using CATIA," *Eng. Appl. Artif. Intell.*, vol. 20, no. 5, pp. 677–690, 2007.

[7] Q. C. Ma and X. W. Liu, "Review of Knowledge Based Engineering with PLM," vol. 12, pp. 127–131, 2008.

[8] D. Pugliese, G. Colombo, and M. Spurio, "About the integration between KBE and PLM," pp. 131–136, 2007.

[9] N. Shintre and A. Shakir, "Knowledge based engineering across Product Realization," no. November, pp. 1–14, 2011.

[10] G. La Rocca, "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design," *Adv. Eng. Informatics*, vol. 26, no. 2, pp. 159–179, 2012.

[11] W. J. C. Verhagen, P. Bermell-Garcia, R. E. C. Van Dijk, and R. Curran, "A critical review of Knowledge-Based Engineering: An identification of research challenges," *Adv. Eng.*

Informatics, vol. 26, no. 1, pp. 5–15, 2012.

[12] G. La Rocca, "Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design," *Adv. Eng. Informatics*, vol. 26, no. 2, pp. 159–179, 2012.

[13] W. Skarka, "Using Knowledge-Based Engineering Methods in Designing with Modular Components of Assembly Systems," pp. 1837–1846, 2010.

[14] C. Landschuetzer, D. Jodin, and A. Wolfschluckner, "Knowledge Based Engineering – an Approach Via Automated Design of Storage / Retrieval Systems," vol. 6, no. 1, pp. 3–10, 2011.

[15] R. E. C. van Dijk, "Intelligent Design Software for a Smart Car," *Leonardo Times, June edition*. pp. 33–36, 2011.

[16] J. Schut, *Conceptual Design Automation*. 2010.

[17] T. Van den Berg, *Harnessing the potential of Knowledge Based Engineering in manufacturing design*. 2013.

[18] C. Verhagen, *An Ontology-Based Approach for Knowledge Lifecycle Management within Aircraft Lifecycle Phases*. 2013.

[19] C. B. Chapman and M. Pinfold, "The application of a knowledge based engineering approach to the rapid design and analysis of an automotive structure," *Adv. Eng. Softw.*, vol. 32, no. 12, pp. 903–912, 2001.

[20] G. Estrada and J. Lloveras, "Application of Moka Methodology To Capture Knowledge in Design for Poka-Yoke Assembly," *Int. Conf. Eng. Des. ICED11*, no. August, pp. 1–10, 2011.

[21] W. Chen, S. Xie, and F. Zeng, "An Integrated Product Development Platform based on STEP and PDM," *Scenario*, no. 3, pp. 1195–1199, 2009.

[22] D. Dhar, S. Computer, and S. Limited, "Integrating Knowledge Based Engineering Product Lifecycle Management."

[23] N. Shintre and A. Shakir, "Knowledge based engineering across Product Realization - Geometric Whitepaper," no. November, pp. 1–14, 2011.

[24] F. Gorski, P. Zawadzki, and A. Hamrol, "Knowledge based engineering as a condition of effective mass production of configurable products by design automation," *J. Mach. Eng.*, vol. 16, no. 4, pp. 5–30, 2016.

[25] J. Forster, I. Arana, and P. Fothergill, "Re-Design Knowledge Representation with

DEKLARE,” in *Proceedings KEML’96 – The Sixth Workshop on Knowledge Engineering: Methods and Languages, Laboratoire de Recherche Informatique, Intelligence Artificialle et Systèmes d’Inférences*, 1996.

[26] P. J. Lovett, A. Ingram, and C. N. Bancroft, “Knowledge-based engineering for SMEs - a methodology,” *J. Mater. Process. Technol.*, vol. 107, no. 1–3, pp. 384–389, 2000.

[27] T. Janis, S. Strong, and J. Wang, “A Methodology for Knowledge Discovery and Classification,” vol. 2, no. 1, pp. 94–103, 2000.

[28] M. Stokes and MOKA Consortium, *Managing Engineering Knowledge : MOKA: Methodology for Knowledge Based Engineering Applications*. London: Professional Engineering, 2001.

[29] R. Curran, W. J. C. Verhagen, M. J. L. Van Tooren, and T. H. Van Der Laan, “A multidisciplinary implementation methodology for knowledge based engineering : KNOMAD,” *Expert Syst. Appl.*, vol. 37, no. 11, pp. 7336–7350, 2010.

[30] A. Mayr *et al.*, “Towards a Knowledge-Based Design Methodology for Managing the Complexity in the Integrated Product and Process Development of Electric Motors,” in *Advances in Production Research*, 2019, pp. 112–125.

[31] Y. L. Karnavas and I. D. Chasiotis, “A simple knowledge base software architecture for industrial electrical machine design: Application to electric vehicle’s in-wheel motor,” *Adv. Intell. Syst. Comput.*, vol. 432, no. September, pp. 111–122, 2016.

[32] C. Favi, M. Germani, M. Marconi, and M. Mengoni, “Innovative software platform for eco-design of efficient electric motors,” *J. Clean. Prod.*, vol. 37, pp. 125–134, 2012.

[33] G. Cross, J. J. Blakley, and F. Shui, “An Expert System for the Integrated Design and Test Result Diagnosis of Industrial DC Motors,” in *International Conference on Electrical Machines;*, 1992, p. p 460-4 vol.2.

[34] C. F. Landy, R. Kaplan, and V. Lun, “An Expert System for the Design of 3-phase squirrel cage Induction Motors,” in *Third International Conference on Electrical Machines and Drives (Conf. Publ. No.282);*, 1987, pp. 127–31.

[35] D. Miljavec, “Set-up an Expert System For the Reluctance Motor Design.” 1998.

[36] C. Zhou, L. Gao, and Z. Liu, “Design and Realize the Electromotor Choice Expert System of Sewing Machine Based on UML,” pp. 1069–1073, 2009.

[37] G. Scutaru, A. Negoita, and R. M. Ionescu, “Three-phase induction motor design software,” *2010 IEEE Int. Conf. Autom. Qual. Testing, Robot.*, vol. 3, pp. 1–4, 2010.

[38] M. A. Jabbar and S. W. Yeo, “A New Architecture of Design Software for Electric

Motors," *Electr. Comput. Eng.*, vol. 1, pp. 360–364, 2000.

[39] J. Laugis and V. Vodovozov, "Expert System for Electric Drive Design," pp. 1017–1019, 2008.

[40] X. Wei, Z. Qiong, and T. Cheng, "Design synthesis of permanent magnet synchronous motor based on expert system," *2010 IEEE Int. Conf. Intell. Comput. Intell. Syst.*, pp. 91–94, 2010.

[41] D. A. Lowther, "Knowledge based systems in electrical machine design," no. 3, pp. 328–333, 1993.

[42] M. V. Deshpande, *Design and Testing of Electrical Machines*. 2010.

[43] J. R. Hendershot, J. R. Hendershot, and T. J. E. Miller, *Design of Brushless Permanent-magnet Machines*. Motor Design Books, 2010.

[44] J. F. Gieras and M. Wing, *Permanent Magnet Motor Technology: design and applications*, vol. 113. 2002.

[45] I. Boldea and S. a. Nasar, *The induction machines design handbook*. 2009.

[46] J. Pyrhonen, T. Jokinen, and V. Hrabovcova, *Design of Rotating Electrical Machines*. 2014.

[47] W. Tong, *Mechanical Design of Electric Motors*. 2014.

[48] P.Beckley, "Electrical steels," pp. 43–80, 2000.

[49] R. Hilzinger and W. Rodewald, *Magnetic Materials*. Wiley, 2013.

[50] G. Lei, J. Zhu, and Y. Guo, *Multidisciplinary Design Optimization Methods for Electrical Machines and Drive Systems*. 2016.

[51] M. Rosu *et al.*, *Multiphysics simulation by design for electrical machines, power electronics, and drives*. Wiley-IEEE Press, 2018.

[52] L. Livadaru, A. Munteanu, A. Simion, and C.-G. Cantemir, "Design and finite element analysis of high-density torque induction motor for traction applications," in *2015 9th International Symposium on Advanced Topics in Electrical Engineering (ATEE)*, 2015, pp. 211–214.

[53] Y. L. Karnavas, I. D. Chasiotis, and S. Member, "A Computational Efficient Algorithm for the Design of a Line Start Synchronous Motor with Multi-Segment Magnet Rotor," p. 93, 2015.

[54] D. G. Dorrell and A. F. F. Filho, "Design of torque-dense induction motors for automotive drive applications," in *2015 IEEE 24th International Symposium on Industrial Electronics*

(ISIE), 2015, pp. 1446–1451.

[55] L. Alberti and N. Bianchi, “Design and tests on a fractional-slot induction machine,” *2012 IEEE Energy Convers. Congr. Expo.*, vol. 2, no. 3, pp. 166–172, 2012.

[56] G. Pellegrino, F. Cupertino, and C. Gerada, “Automatic Design of Synchronous Reluctance Motors focusing on Barrier Shape Optimization,” *IEEE Trans. Ind. Appl.*, 2014.

[57] Y. Xie, H. Ge, Z. Yang, and S. Lv, “Study on key technologies of permanent magnet synchronous motor design for electric vehicles,” *2014 17th Int. Conf. Electr. Mach. Syst.*, pp. 2158–2163, 2014.

[58] J. W. Jansen, J. P. C. Smeets, T. T. Overboom, J. M. M. Rovers, and E. A. Lomonova, “Overview of Analytical Models for the Design of Linear and Planar Motors,” *IEEE Trans. Magn.*, vol. 50, no. 11, pp. 1–7, 2014.

[59] P. Ponmurugan, “Multiobjective Optimization of Electrical Machine , a State of the Art Study,” vol. 56, no. 13, pp. 26–30, 2012.

[60] G. Lei, C. Liu, J. Zhu, and Y. Guo, “Techniques for Multilevel Design Optimization of Permanent Magnet Motors,” *IEEE Trans. Energy Convers.*, vol. PP, no. 99, pp. 1–11, 2015.

[61] G. Bramerdorfer, S. Silber, G. Weidenholzer, and W. Amrhein, “Comprehensive cost optimization study of high-efficiency brushless synchronous machines,” in *Electric Machines & Drives Conference (IEMDC), 2013 IEEE International*, 2013, pp. 1126–1131.

[62] M. Centner, “Basics and application of motor design optimization in an industrial environment,” *2014 Int. Conf. Electr. Mach.*, pp. 1008–1012, 2014.

[63] S. Ghosh, H. Memane, M. Karpe, R. Birajdar, V. K. Yadav, and V. Mukherjee, “Design of high performance induction motor through multi objective and multi domain optimization,” in *2015 5th International Electric Drives Production Conference (EDPC)*, 2015, pp. 1–6.

[64] J. A. Malagoli, J. R. Camacho, S. Member, M. V. F. Luz, J. H. I. Ferreira, and A. M. Sobrinho, “Design of Three-Phase Induction Machine Using Differential Evolution Algorithm,” vol. 13, no. 7, pp. 2202–2208, 2015.

[65] INCOSE, “Systems engineering handbook - A guide for system life cycle processes and activities,” p. 382, 2010.

[66] K. Forsberg and H. Mooz, “The Relationship of System Engineering to the Project Cycle,”

12th INTERNET World Congr. Proj. Manag., no. June 1994, p. 12, 1994.

[67] A. T. Bahill and B. Gissing, “Re-evaluating Systems Engineering Concepts Using Systems Thinking,” *IEEE Trans. Syst. Man, Cybern. C Appl. Rev.*, vol. 28, no. 4, pp. 516–527, 1998.

[68] G. Chang, H. Perng, and J. Juang, “A review of systems engineering standards and processes,” *J. Biomechatronics Eng.*, vol. 1, no. 1, pp. 71–85, 2008.

[69] “OMG SysML,” 2016. [Online]. Available: http://www.omgsysml.org/what-is-sysml.htm. [Accessed: 01-Nov-2016].

[70] D. Dori, *Model-Based Systems Engineering with OPM and SysML*. 2016.

[71] T. Weilkiens, *Systems Engineering with SysML/UML*. 2007.

[72] S. Distilled *et al.*, “Praise for SysML Distilled.”

[73] T. Systems and M. Language, *A Practical Guide to SysML*. .

[74] J. Murray, “Model Based Systems Engineering (MBSE) Media Study,” *Int. Counc. Syst. Eng.*, 2012.

[75] G. Pahl, W. Beitz, J. Feldhusen, and K.-H. Grote, *Engineering design: a Systematic Approach*. 2007.

[76] N. Cross, *Engineering design methods*. 2008.

[77] V. D. Ingenieure, “VDI 2221 Systematic Approach to the Design of Technical Systems and Products.” 1993.

[78] V. S. Vasi, “Standard Industrial Guideline for Mechatronic Product Design,” pp. 103–108, 2008.

[79] R. Aydin, C. K. Kwong, and P. Ji, “A novel methodology for simultaneous consideration of remanufactured and new products in product line design,” *Int. J. Prod. Econ.*, vol. 169, pp. 127–140, 2015.

[80] R. Nattermann and R. Anderl, “The W-model using systems engineering for adaptronics,” *Procedia Comput. Sci.*, vol. 16, no. 0, pp. 937–946, 2013.

[81] K. M. Adams, *Nonfunctional Requirements in Systems Analysis and Design*, vol. 28. Cham: Springer International Publishing, 2015.

[82] K. Gericke, “COMPARISONS OF DESIGN METHODOLOGIES AND PROCESS MODELS ACROSS DISCIPLINES : A LITERATURE REVIEW,” no. August, 2011.

[83] T. Tomiyama, P. Gu, Y. Jin, D. Lutters, C. Kind, and F. Kimura, “Design methodologies: Industrial and educational applications,” *CIRP Ann. - Manuf. Technol.*, vol. 58, no. 2, pp.

543–565, 2009.

[84] D. Motte, “Effectiveness of the systematic engineering design methodology,” in *Proceedings of the 20th International Conference on Engineering Design - ICED’15*, 2015.

[85] “VDI 2206 Design methodology for mechatronic systems.” 2004.

[86] A. R. Ramhuzaini Bin, U. Pulm, and R. Stetter, “Systematic mechatronic design of a piezo-electric brake,” *Int. Conf. Eng. Des. ICED’07*, no. August, pp. 1–12, 2007.

[87] M. Follmer, P. Hehenberger, S. Punz, and K. Zeman, “Approach for the creation of mechatronic system models,” in *International Conference on Engineering Design, ICED11*, 2011, no. August.

[88] M. Eigner and T. Gilz, “Interdisciplinary Product Development - Model Based Systems Engineering Radoslav Zafirov,” 2013.

[89] J. Jänsch and H. Birkhofer, “The development of the guideline VDI 2221 - the change of direction,” *Int. Des. Conf. - Des. 2006*, pp. 45–52, 2006.

[90] J. Gausemeier and S. Moehringer, “NEW GUIDELINE VDI 2206 – A FLEXIBLE PROCEDURE MODEL FOR Specific Requirements to the Design of Mechatronic Systems,” *Int. Conf. Eng. Des. Iced 03 Stock.*, p. 10, 2003.

[91] I. Gomez, “Design Methodology for Achieving Reliable Permanent Magnet Synchronous Machines,” Mondragon Unibertsitatea, 2017.

[92] G. Schuh, J. Arnoscht, and S. Rudolf, “Integrated development of modular product platforms,” *PICMET ’10 - Portl. Int. Cent. Manag. Eng. Technol. Proc. - Technol. Manag. Glob. Econ. Growth*, pp. 1928–1940, 2010.

[93] A. T. Bahill and S. J. Henderson, “Requirements Development, Verification, and Validation Exhibited in Famous Failures,” *Syst. Eng.*, vol. 8, no. 1, pp. 1–14, 2005.

[94] ABB, “Low voltage motors Motor guide.” ABB, 2018.

[95] WEG, “Specification of Electric Motors.” WEG Group, 2016.

[96] T. Jones, “Motor Efficiency , Selection, and Management A Guidebook for Industrial Efficiency Programs,” no. September. Consortion for Energy Efficiency, 2013.

[97] A. Chausovsky, “‘Motor Market Update’, Presentation Slides, Motor Summit.” Motor Summit 2014, Zurich, 2014.

[98] J. Dick, E. Hull, and K. Jackson, *Requirements Engineering*. Springer, 2017.

[99] Y. Arayici, V. Ahmed, and G. Aouad, “A Requirements Engineering Framework for

Integrated Systems Development for the Construction Industry," *J. Inf. Technol. Constr.*, vol. 11, no. October 2005, pp. 35–55, 2006.

[100] S. Rehman and V. Gruhn, "An Effective Security Requirements Engineering Framework for Cyber-Physical Systems," *Technologies*, vol. 6, no. 3, p. 65, 2018.

[101] R. Abo, "Introduction to a Requirements Engineering Framework for Aeronautics," *J. Softw. Eng. Appl.*, vol. 03, no. 09, pp. 894–900, 2010.

[102] ABB, "Optimizer." [Online]. Available: http://www145.abb.com/selection. [Accessed: 14-Nov-2018].

[103] A. Kampker, K. Kreisköther, M. K. Büning, P. Treichel, and J. Theelen, "Automotive quality requirements and process capability in the production of electric motors," 2017.

[104] M. Dufrene, "A methodological framework to support integrated ecodesign for companies : requirements and conceptualization towards a software platform.," University Grenoble Alpes, 2015.

[105] M. Jufer, *Electric Drives: Design Methodology*. Wiley-ISTE press, 2013.

[106] M. Yilmaz, "Limitations/capabilities of electric machine technologies and modeling approaches for electric motor design and analysis in plug-in electric vehicle applications," *Renew. Sustain. Energy Rev.*, vol. 52, pp. 80–99, 2015.

[107] G. A. Mccoy and J. G. Douglass, "Premium efficiency motor selection and application guide - A Handbook for Industry." U.S. Department of Energy, 2014.

[108] R. B. Boteler, M. E. Moore, and M. J. Turner, "An evaluation and comparison of advanced motor technologies," in *9th International Conferenece on Energy Efficiency in Motor Driven Systems EEMODS 2015*, 2016, pp. 412–423.

[109] T. Luft, D. Roth, H. Binz, and S. Wartzack, "A New 'Knowledge-Based Engineering' Guideline," *Inte*, vol. 6, no. August, pp. 207–216, 2017.

[110] S. L. Kendal and M. Creen, *An Introduction to Knowledge Engineering*. 2007.

[111] D. Sriram, G. Stephanopoulos, and R. Logcher, "Knowledge-based system applications in engineering design: Research at MIT," *AI Mag.*, vol. 10, no. 3, pp. 79–96, 1989.

[112] J. Durkin, "Research Review : Application of Expert Systems in the Sciences," *Ohio J. Sci.*, vol. 90, no. 5, pp. 171–179, 1990.

[113] P. Lucas and L. van der Gaag, "Principles of Expert Systems," 1991.

[114] M. D. Rychener, "Expert systems for engineering design : problem components , techniques , and prototypes," 1983.

[115] D. Redington, “A review of knowledge engineering and expert systems: Towards Expert Operators in Forth.” 1985.

[116] J. Sauer, “Knowledge-Based Systems Techniques and Applications in Scheduling,” *Knowledge-Based Syst.*, vol. 231, no. 4741, p. 1554, 2000.

[117] D. Poole and A. Mackworth, *Artificial Intelligence Foundations of Computational Agents*, Second edi. Cambridge University Press, 2017.

[118] P.-H. Speel, a. T. Schreiber, W. Van Joolingen, G. Van Heijst, and G. J. Beijer, “Conceptual modelling for knowledge-based systems,” *Encycl. Comput. Sci. Technol.*, pp. 1–24, 2001.

[119] P. S. Sajja and R. Akerkar, “Knowledge-Based Systems for Development,” *Adv. Knowl. Based Syst. Model. Appl. Res.*, vol. 1, pp. 1–11, 2010.

[120] R. Studer, V. R. Benjamins, and D. Fensel, “Knowledge engineering: Principles and methods,” *Data Knowl. Eng.*, vol. 25, pp. 161–197, 1998.

[121] C. L. Emberey, N. R. Milton, J. P. T. J. Berends, M. J. L. Van Tooren, S. W. G. Van Der Elst, and B. Vermeulen, “Application of Knowledge Engineering Methodologies to Support Engineering Design Application Development in Aerospace,” *7th AIAA Aviat. Technol. Integr. Oper. Conf.*, no. September, pp. 18–20, 2007.

[122] K.-K. Lau, *Component-Based Software Development: Case Studies (Series on Component-Based Software Development)*. World Scientific Press, 2004.

[123] W. Hasselbring, “Component-based software engineering,” in *Handbook of Software Engineering and Knowledge Engineering*, no. May 2002, C. S.K., Ed. Singapore: World Scientific Publishing, 2002, pp. 289–305.

[124] I. Sommerville, *Software Engineering*. 2016.

[125] C. Szyperski, *Component Software*, Second edi. Addison-Wesley, 2002.

[126] G. T. Heineman and W. T. Councill, *Component-Based Software Engineering: Putting the Pieces Together*. Addison-Wesley, 2001.

[127] J. Stark, *Product Lifecycle Management (Volume 1)*, vol. 1. 2015.

[128] G. Schuh, H. Rozenfeld, D. Assmus, and E. Zancul, “Process oriented framework to support PLM implementation,” *Comput. Ind.*, vol. 59, no. 2–3, pp. 210–218, 2008.

[129] SolidWorks, “PDM VS. PLM: It all starts with PDM,” 2010.

[130] M. Huhtala, M. Lohtander, and J. Varis, “Confusing of terms PDM and PLM : examining issues from the PDM point of view .,” no. November 2016, 2012.

[131] J. Stark, *Product Lifecycle Management (Volume 2)*, vol. 2. 2016.

[132] A. Nguyen-Duc and P. Abrahamsson, “Minimum Viable Product or Multiple Facet Product? The Role of MVP in Software Startups,” in *Proceedings of the International Conference on Agile Software Development*, 2016, pp. 118–130.

[133] M. Hirz, W. Dietrich, A. Gfrerrer, and J. Lang, *Integrated Computer-Aided Design in Automotive Development*. Springer, 2013.

[134] Evangelos Triantaphyllou, *Multi-criteria Decision Making Methods*. Springer US, 2000.

[135] M. Velasquez and P. T. Hester, “An Analysis of Multi-Criteria Decision Making Methods,” 2013.

[136] E. Ries, *The Lean Startup: How Today’s Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses Eric Ries. New York: Crown Business, 2011. 320 pages. US$26.00.* New York: Crown Business, 2011.

[137] “SyncDev Process, Tools, and Services,” 2008. [Online]. Available: http://web.archive.org/web/20081003054508/http://www.productdevelopment.com/howitworks/mvp.html. [Accessed: 28-Aug-2019].

[138] E. Anderson, S. Y. Lim, and N. Joglekar, “Are More Frequent Releases Always Better? Dynamics of Pivoting, Scaling, and the Minimum Viable Product,” *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, pp. 5849–5858, 2017.

[139] D. Rancic Moogk, “Minimum Viable Product and the Importance of Experimentation in Technology Startups,” *Technol. Innov. Manag. Rev.*, vol. 2, no. 3, pp. 23–26, 2012.

[140] V. Lenarduzzi and D. Taibi, “MVP Explained: A Systematic Mapping Study on the Definitions of Minimal Viable Product,” *Proc. - 42nd Euromicro Conf. Softw. Eng. Adv. Appl. SEAA 2016*, pp. 112–119, 2016.

[141] S. Armentia, S. M. De Pancorbo, G. Ugalde, J. Poza, and A. J. Escalada, “Experimental Validation of FEM Based Design Method for Synchronous Reluctance Machine,” in *XXII International Conference on Electrical Machines (ICEM)*, 2016, pp. 1716–1722.

[142] M. Asimov, *Introduction to design.* Englewood Cliffs, N.J: Prentice-Hall., 1962.

[143] M. J. French, *Conceptual design for engineers (3rd edition)*. Springer, 1998.

[144] V. Hubka and W. Eder, *Principles of engineering design*. Butterworth & Co. Ltd., 1982.

APPENDIX

This section presents the appendices of the dissertation.

Appendix A **Minimum viable product (MVP)**

The term minimum viable product (MVP) according to Ries [136] p.77 is a product version “that enables a full turn of the Build-Measure-Learn loop with a minimum amount of effort and the least amount of development time”. Robinson [137] says that MVP “is the right-sized product for your company and your customer. It is big enough to cause adoption, satisfaction, and sales, but no so big as to be bloated and risky. Technically, it is the product with maximum ROI divided by risk”. The interest is put in the technical features of this term and not in the economical ones.

The aim of this review is to introduce MVP because later in this work it is suggested that the devised KBE application be developed following this principle. One of the main advantage of introducing this term in this work is to get feedback from the end-user in order to make further decisions about the KBE application. The author is seeking ways to enhance the software development time of KBE applications. The technical uncertainty in KBE application is low since its starting point is well-stablished knowledge, so what the client wants is known, the problem relies in how useful and how much will the KBE application enhance the product development process disregarding the knowledge retention.

Anderson et al. [138] summarizes the objectives of MVP as maximizing the learning and minimizing uncertainty with the minimum resources. MVP is closely related to other software engineering terms such as rapid prototyping and agile development methods. The commonality resides in developing software in multiple facets for faster deployment by eliminating wasted development time or other resources. Implementing MVP engages the target customers and tests the value of the software [139].

Lenarduzzi and Taibi [140] collect several definitions of MVP, additionally they summarize the main purposes of a MVP. From which the author would like to highlight the MVP with minimum functionalities, which have as main purpose rapid deployment, gather user feedback, and test the fundamental hypothesis of the business. These features target the objective the author is looking for with the creation of a KBE application. From the classification of MVP types [132] the rip-off is of major interest because is a software with the main features to get feedback to later pivot in different directions if necessary.

Appendix B **eMachine algorithms and validation**

The proposed algorithms presents an alternative of automatic computation of brushless motors operation points within the same FEM tool and within acceptable times of execution by iterative processes. From this point forward τ_{emf} is the electromagnetic torque computed with the dq fluxes. Furthermore, V_t is the line-to-line voltage. All the results for a particular current coordinate are called characteristic variables which is illustrated in Fig. A-1 and depending the context of study the phrase characteristic variables include all or the ones that are relevant in the context.

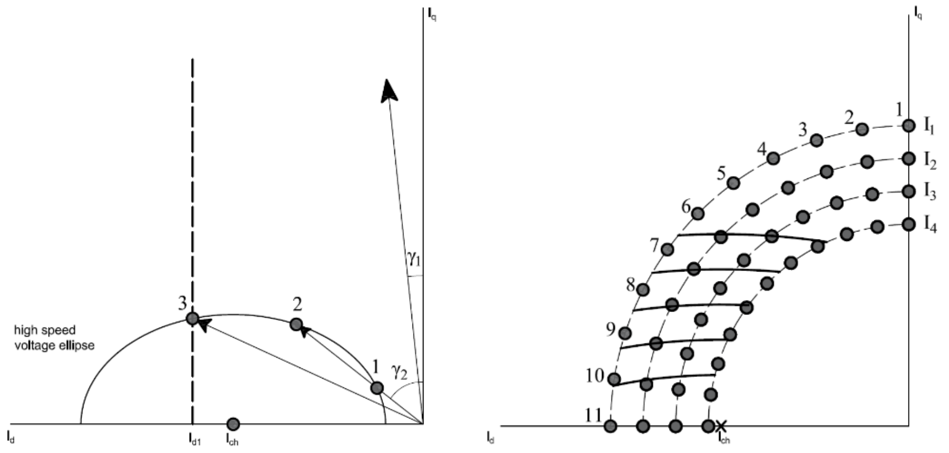
$$\left.\begin{aligned} \vec{I}_a &= \sqrt{2}I_s\angle\gamma \\ \vec{I}_b &= \sqrt{2}I_s\angle\gamma-120^\circ \\ \vec{I}_c &= \sqrt{2}I_s\angle\gamma+120^\circ \end{aligned}\right\} \rightarrow \boxed{\boxed{\text{FEM tool}}} \rightarrow \left\{\begin{bmatrix} \vec{\psi}_a \\ \vec{\psi}_b \\ \vec{\psi}_c \\ \tau_{em} \end{bmatrix}\right\}$$

$$\text{characteristic variables} \rightarrow \left[I_s,\gamma,\psi_{ds},\psi_{qs},\tau_{emf},V_t,N_{rpm}\right]$$

Fig. A-1. FEM tool inputs and outputs, and characteristic variables

Algorithms

The objective of Algorithm 1 is to determine the targeted current magnitude for a given torque and a fixed angle value, in this case the secant method is used to converge to the result, and usually it finds the value in less than ten iterations if the first guessed is not too far. In a similar way using the secant method, Algorithm 2's objective is to determine the characteristic variables for a given voltage value for a fixed I_d. In this case, the angle γ is parametrized with a fixed I_d current to avoid divergence of the algorithm. This is better explained with Fig. A-2a, if the voltage ellipse is small i.e., for a high speed, the algorithm may diverge if the angle is not correctly targeted, for example for γ_1 there is no convergence to the given voltage and speed, for γ_2 depending on the initial guess the algorithm may converge to point 1 or 2. Therefore, in order to have a sort of control an I_d current is fixed, having iterations only in I_q until convergence e.g., point 3, otherwise we are outside of the voltage ellipse. For both algorithms to compute the derivative approximation in the secant method two first guesses are needed in this case zero was given as one initial value for the torque and current since it is a known point, however, this may be changed or the algorithm may be adjusted to compute two initial guesses



(a) (b)

Fig. A-2. a) Voltage ellipse computation with algorithm 2; b) Sweep of currents in algorithm 6 and 9.

Using the secant method in Algorithm 3 we get I_{ch} based on Equation (1) when solved for I_{d}.

$$\begin{pmatrix}\psi_{ds}\\ \psi_{qs}\end{pmatrix}=\begin{bmatrix}L_{ds} & L_m\\ L_m & L_{qs}\end{bmatrix}\begin{pmatrix}I_{ds}\\ I_{qs}\end{pmatrix}+\begin{pmatrix}\psi_{pm}^{d}\\ \psi_{pm}^{q}\end{pmatrix} \qquad (1)$$

I_{ch} is the value of I_{d} needed to drive ψ_{ds} to zero. An initial value of one was given to flux d to avoid zero division error, this can be changed by computing two initial guesses or give the magnet flux if known for the PMSMs. In Algorithm 4 we target the value of the MAPT point for a given torque, in summary what is done is to compute the constant torque curve for different angles using Algorithm 1, then using cubic spline interpolation obtain the minimum current magnitude. The MAPT point is basically the same MTPA point but targeted with the torque instead of the current.

Algorithm 1: Given the torque and angle get the current magnitude that produces that torque

```
Given $\tau_{ref}$, $\gamma_{ref}$, tolerance $\varepsilon_{\tau_{ref}} > 0$, k: = 0, kmax, initial $I_{s_0}$, step_number = n, motor model and tool parameters
Initialize variables $I_s$ = [0]; $\tau$ = [0]; converge = False
While not converge:
    simulate point ($I_{s_k}$, $\gamma_{ref}$) with given step_number
    get $\overrightarrow{\psi_a}, \overrightarrow{\psi_b}, \overrightarrow{\psi_c}$
    compute $\psi_{ds}$, $\psi_{qs}$
    compute $\tau_k$ with Equation (3);
```

```
    add values to $I_s$ and $\tau$
    If $\left|\tau_{ref} - \tau_k\right| \le \varepsilon_{\tau_{ref}}$ :
        converge = True
        return(characteristic variables)
    elif k = kmax: break
    else:
        $\Delta I_s = I_{s_k} - I_{s_{k-1}}$ ; $\Delta \tau = \tau_k - \tau_{k-1}$
        $I_{s_{k+1}} = I_{s_k} + \frac{\Delta I_s}{\Delta \tau}\left(\tau_{ref} - \tau_k\right)$
    k = k + 1
```

Algorithm 2: Given the speed, voltage and Id current get the characteristic variables where the voltage is reached

```
Given $N_{rpm}$, $V_{ref}$ , tolerance $\varepsilon_{V_{ref}} > 0$, $I_d$, k: = 0, kmax, initial $I_{q_0}$, step_number=n, model and tool parameters
Initialize variables $I_q$ = [0]; $V_t$ = [0]; converge = False
While not converge:
    compute $I_{S_k}$ with Equation (4)
    compute $\gamma_k$ with atan($I_d / I_{q_k}$)
    simulate point ($I_{S_k}$, $\gamma_k$)
    get $\overrightarrow{\psi_a}, \overrightarrow{\psi_b}, \overrightarrow{\psi_c}$ and compute $\psi_{ds}$ , $\psi_{qs}$
    compute $V_k$ with Equation (1)
    append values to $V_t$ and $I_q$
    If $\left|V_{ref} - V_k\right| \le \varepsilon_{V_{ref}}$ :
        converge=True; compute $\tau_{emf}$
        return(characteristic variables)
    elif k = kmax: break
    else:
        $\Delta V_t = V_k - V_{k-1}$ ; $\Delta I_q = I_{q_k} - I_{q_{k-1}}$
        $I_{q_{k+1}} = I_{q_k} + \frac{\Delta I_q}{\Delta V_t}\left(V_{ref} - V_k\right)$
    k = k + 1
```

Algorithm 3: Get Ich of the motor

```
Given tolerance $\varepsilon_{\psi_{ds}}$ > 0, initial $I_{s_0}$, k: = 0, kmax, step_number = n, model and tool parameters
Initialize variables $I_s$ = [0]; $\tau$ = [0]; $\psi_d$ = [1]; converge = False
While not converge:
    simulate point ( $I_{s_k}$, 90)
    get $\vec{\psi}_a, \vec{\psi}_b, \vec{\psi}_c$ and compute $\psi_{d_k}$ append results in $I_s$, $\tau$, $\psi_d$
    If $\left|\psi_{d_k}\right| \le \varepsilon_{\psi_{ds}}$ :
        converge=True; compute $\tau_{emf}$
        return(characteristic variables)
    elif k=kmax: break
    else:
        $\Delta\psi_d = \psi_{d_k} - \psi_{d_{k-1}}$; $\Delta I_s = I_{s_k} - I_{s_{k-1}}$
        $I_{s_{k+1}} = I_{s_k} + \frac{\Delta I_s}{\Delta\psi_d}\left(-\psi_{d_k}\right)$
    k = k + 1
```

Algorithm 4: Given the torque get the MAPT point

```
Given $\tau_{ref}$, tolerance $\varepsilon_{\tau_{ref}}$ > 0, $N_{rpm}$, $\vec{\gamma} = [\gamma_0, \gamma_1, ..., \gamma_{end}]$ where $\gamma_0 < \gamma_{end}$, initial $I_{s_0}$, k: = 0, step_number = n, motor model and tool parameters
Initialize variables $I_s$ = []; $\tau$ = []; $\psi_d$ = []; $\psi_q$ = []; $\gamma$ = [];
For each value in $\vec{\gamma}$ run Algorithm 1 with $I_{s_k}$
Append results in $I_s$, $\tau$, $\psi_d$, $\psi_q$, $\gamma$
Stop when $I_{s_k} > I_{s_{k-1}}$
Optional. After stop apply bisection as needed between $\gamma_k$ and $\gamma_{k-1}$ for a better approximation, particularly when steps in $\vec{\gamma}$ are large.
Apply cubic spline interpolation to variables $I_s$, $\tau$, $\psi_d$, $\psi_q$, $\gamma$
Get min( $I_s$ ) and its index
Get the rest of variables at that index.
```

```
Compute $V_t$ with Equation (1)
return(characteristic variables)
```

In Algorithm 5 we target the MTPA point for a given current, this is a simple process of sweeping different angles for the same current magnitude, then using cubic spline interpolation obtain the maximum torque value and return the characteristic variables for that point. Algorithm 6 uses the Newton's method to obtain the speed for a given voltage limit, in a specific current vector, the process is illustrated in Fig. A-2b, for a current I_1 and a given set of angles 1 to 11 in the figure the algorithm finds the corresponding speed for the voltage limit for each point. In this case we need to compute the derivative of the voltage with respect to speed, this derivative is given in Equation (2). The MTPV point for a given speed and voltage limit is obtained through Algorithm 7, it uses Algorithm 2 for different I_d currents getting values along the voltage ellipse and finally uses cubic spline interpolation to find the maximum torque.

$$\frac{dV_t}{dw_s} = \frac{\sqrt{3}\left[\psi_{ds}\left(w_s\psi_{ds} + R_s I_{qs}\right) + \psi_{qs}\left(w_s\psi_{qs} - R_s I_{ds}\right)\right]}{\sqrt{\left(w_s\psi_{ds} + R_s I_{qs}\right)^2 + \left(w_s\psi_{qs} - R_s I_{ds}\right)^2}} \quad (2)$$

Algorithm 5: Given the current limit get the MTPA point

```
Given $I_{s_{ref}}$, $N_{rpm}$, $\vec{\gamma}=[\gamma_0,\gamma_1,\ldots,\gamma_{end}]$ where $\gamma_0<\gamma_{end}$, k: = 0, step_number = n, motor model and tool parameters
Initialize variables $\tau$ = []; $\psi_d$ = []; $\psi_q$ = []; $\gamma$ = [];
For each value in $\vec{\gamma}$ simulate point($I_{s_{ref}}$, $\gamma_k$), get $\vec{\psi}_a,\vec{\psi}_b,\vec{\psi}_c$ and compute $\psi_{ds}$, $\psi_{qs}$, $\tau_{emf}$
Append results in $\tau$, $\psi_d$, $\psi_q$, $\gamma$
Stop when $\tau_k < \tau_{k-1}$
Optional. After stop apply bisection as needed between $\gamma_k$ and $\gamma_{k-1}$ for a better approximation, particularly when steps in $\vec{\gamma}$ are large.
Apply cubic spline interpolation to variables $\tau$, $\psi_d$, $\psi_q$, $\gamma$
Get max($\tau$) and its index
Get the rest of variables at that index
Compute $V_t$ with Equation (1)
return(characteristic variables)
```

Algorithm 6: Given the current and voltage limit get the FW speeds along the current circle

```
Given $I_{s_{ref}}$, $V_{ref}$, tolerance $\varepsilon_{V_{ref}} > 0$, initial $w_0$, $\vec{\gamma} = [\gamma_0, \gamma_1, ..., \gamma_{end}]$ where $\gamma_0 < \gamma_{end}$, k:= 0, kmax, step_number = n, motor model and tool parameters
Initialize variables $\tau$ = []; $\psi_d$ = []; $\psi_q$ = []; converge = False
For each value in $\vec{\gamma}$ simulate point ( $I_{s_{ref}}$, $\gamma_k$ ), get $\vec{\psi}_a, \vec{\psi}_b, \vec{\psi}_c$ and compute $\psi_{ds}$, $\psi_{qs}$, $\tau_{emf}$
Append results in $\tau$, $\psi_d$, $\psi_q$
  For each item index i in $\vec{\gamma}$ :
    While not converge:
      compute $V_k$ with $w_k$, $\psi_d$[i], $\psi_q$[i], $\vec{\gamma}$[i] in Equation (4)
      If $\left|V_{ref} - V_k\right| \le \varepsilon_{V_{ref}}$ :
            converge = True; compute $N_k$;
            return(characteristic variables)
      elif k = kmax: break;
      else:
            compute $\frac{dV_k}{dw_k}$ with Equation (8)
            $w_{k+1} = w_k + \left(\frac{dV_k}{dw_k}\right)^{-1}\left(V_{ref} - V_k\right)$
    k = k + 1
```

Algorithm 8 and 9 objective is to find mode 1 and 2 limit, respectively. Algorithm 8 assumes the limit is given by the voltage, it iterates along the MTPA curve using Algorithm 5 until the voltage is found as illustrated in Figure 1, in this case we do not take into account the current or torque limit, the user criteria must decide if the point is valid or not. This was done because if the torque or current decide the limit then Algorithm 4 or Algorithm 5 can be used. Finally, Algorithm 9 finds the mode 2 limit, the CPSR and the MPSR point for the case where $I_{\mathrm{ch}} < I_{\max}$ because for the case where $I_{\mathrm{ch}} \geq I_{\max}$ a simpler analysis using Algorithm 6 can be done. These points can be found in different ways e.g., using Algorithm 7, however, its convergence is slower. Therefore, we use Algorithm 6 due to FEM tools simulate a working point targeted with phase currents, then, converging faster compared to any other way of iteration. The process can be explained with Fig. A-2b, here we use Algorithm 6 for different currents I_1 to I_4 in this example, and then we automatically process the results using cubic spline interpolation to find different query speeds voltage ellipses (the solid lines crossing the currents) then applying interpolation to find the maximum torque on each ellipse, to find finally the limits of the motor.

Algorithm 7: Given the speed and voltage limit what is the MTPV on the corresponding voltage ellipse

```
Given $N_{ref}$, $V_{ref}$, tolerance $\varepsilon_{V_{ref}} > 0$, $\overrightarrow{I_d} = [I_{ch}, I_{d1}, \ldots, I_{dend}]$ where $I_{ch} < I_{dend}$, k: = 0, kmax, step_number = n, motor model and tool parameters
Initialize variables $\tau$ = []; $I_s$ = [], $\psi_d$ = []; $\psi_q$ = []; $\gamma$ = []; Stop = False
While not Stop:
    for each value in $\overrightarrow{I_d}$ run Algorithm 2
        append results in $\tau$, $I_s$, $\psi_d$, $\psi_q$, $\gamma$
        If $\tau_k < \tau_{k-1}$ : Stop=True; Optional. After stop apply bisection as needed between $I_{dk}$ and $I_{dk-1}$ for a better approximation, particularly when steps in $\overrightarrow{I_d}$ are large.
Apply cubic spline interpolation to variables $\tau$, $I_s$, $\psi_d$, $\psi_q$, $\gamma$
Get max($\tau$) and its index
Get the rest of variables at that index
return(characteristic variables)
```

Algorithm 8: Given the voltage limit get mode 1 limit.

```
Given $V_{ref}$, tolerance $\varepsilon_{V_{ref}} > 0$, $\vec{\gamma} = [\gamma_0, \gamma_1, \ldots, \gamma_{end}]$ where $\gamma_0 < \gamma_{end}$, initial $I_{s_0}$ and $I_{s_1}$, $N_{rpm}$, k: = 0, kmax, step_number = n, motor model and tool parameters
Initialize variables $\tau$ = []; $I_s$ = [], $\psi_d$ = []; $\psi_q$ = []; $\gamma$ = []; $V_t$ = []; Stop = False
Run Algorithm 5 for $I_{s_0}$ and $I_{s_1}$
Append results in $\tau$, $I_s$, $\psi_d$, $\psi_q$, $\gamma$, $V_t$, stop if $\left|V_{ref} - V_k\right| \le \varepsilon_{V_{ref}}$
While not Stop:
    $\Delta V_t = V_k - V_{k-1}$; $\Delta I_s = I_{s_k} - I_{s_{k-1}}$
    $I_{s_{k+1}} = I_{s_k} + \frac{\Delta I_s}{\Delta V_t}\left(V_{ref} - V_k\right)$
    run Algorithm 5 append results in $\tau$, $I_s$, $\psi_d$, $\psi_q$, $\gamma$, $V_t$
    If $\left|V_{ref} - V_k\right| \le \varepsilon_{V_{ref}}$ :
        Stop=True
        return(characteristic variables)
    elif k = kmax: break
    k = k + 1
```

Algorithm 9: Given the voltage and current limit get the mode 2 limit, CPSR and MPSR.

```
Given $V_{t\max}$, $I_{s\max}$, $\gamma_{MTPA}$, $\vec{\gamma}=[\gamma_0,\gamma_1,...,\gamma_{end}]$ where $\gamma_0<\gamma_{end}$ and $\gamma_0=\gamma_{MTPA}$, $I_{ch}$, $N_{rated}$, k: = 0, $k_{\max}$, step_number = n, motor model and tool parameters.
Initialize variables $\tau$ = []; $N_{rpm}$ = []; $\tau_{interp}$ = []; $\gamma_{interp}$ = []; $N_{interp}$ = []; $\tau_N$ = []; $\gamma_N$ = []; $\tau_{mtpv}$ = []; $\gamma_{mtpv}$ = []; $I_{smtpv}$ = [], $P_{em}$ = []
create array $\overrightarrow{I_{s\_v}}=[I_{s\max},I_{s1},...,I_{send}]$ where $I_{s\max}>I_{send}>I_{ch}$
for each value in $\overrightarrow{I_{s\_v}}$ and for each $\vec{\gamma}$ run Algorithm 6 append results in $\tau$, $N_{rpm}$
Apply cubic spline interpolation to $\vec{\tau},\vec{\gamma},N_{rpm}$ with more $\vec{\gamma}$ steps and append in $\tau_{interp}$, $\gamma_{interp}$, $N_{interp}$
create array $\overrightarrow{N_{query}}=[N_0,N_1,...,N_{end}]$ where $N_{rated}<N_0<N_{end}$
For each value in $\overrightarrow{N_{query}}$ get vectors $\tau_N$, $\gamma_N$ for each $\overrightarrow{I_{s\_v}}$ by cubic spline interpolation using $\tau_{interp}$, $\gamma_{interp}$, $N_{interp}$, $\overrightarrow{I_{s\_v}}$
For each speed in $\overrightarrow{N_{query}}$ interpolate $\tau_N,\gamma_N$, $\overrightarrow{I_{s\_v}}$ with more $\gamma_N$ steps
        get max($\tau_N$) and its index
        get $\gamma_N$, $\overrightarrow{I_{s\_v}}$, value at that index and append results in $\tau_{mtpv}$, $\gamma_{mtpv}$, $I_{smtpv}$
        Compute power with $\tau_{mtpv}$ and $\overrightarrow{N_{query}}$ append results in $P_{em}$
With $P_{em}$, $\tau_{mtpv}$, $\gamma_{mtpv}$, $I_{smtpv}$, $\overrightarrow{N_{query}}$ get mode 2 limit characteristic variables which is the highest speed at $I_{s\max}$, get CPSR point and characteristic variables which is the highest speed where $P_{em}$ is equal to $P_{rated}$, if no value found then CPSR is infinite and get the MPSR point with the speed at the maximum point in $P_{em}$.
return (characteristic variables for mode 2 limit, CPSR and MPSR)
```

Validation

To validate the algorithms the speed characteristics for the three types of brushless AC motors are computed, the motors here mentioned have already validated design models [141]. General characteristics of the motors are mentioned in Table A-1, for the purpose of this section the designs were intentionally modified in order to have their characteristic current (I_{ch}) below rated currents. The algorithms were developed in Jython and are implemented in the FEM tool Flux 2D/3D of Altair. The computer used is an intel core i5-4590 CPU with 16 GB of RAM and an OS Windows 7–64 bits.

The process here undertaken consist of computing the three limits of operation for each motor, some points on the MTPA, FW and MTPV curves, the CPSR and MPSR point, then, the results are plotted in order to show the circle diagram and speed characteristics for each motor. The mode 1, 2 and 3 limits are compared with its reference value and the time of execution in seconds of each algorithm are given, although this can vary depending the inputs of the algorithms and different factors of the motor model design. First, for all cases the model in the FEM tool is set to be control in polar mode, the simulations are set to be done in one-sixth of an electrical period in magnetostatic mode.

Table A-1. Motor characteristics

Parameter	Non-Salient PM	IPM	SynRM
Maximum voltage [V]	400	400	400
Power [kW]	13	7	6
Rated Speed [RPM]	166	166	166
Pole pair	10	3	3

Selection of the Step Number:

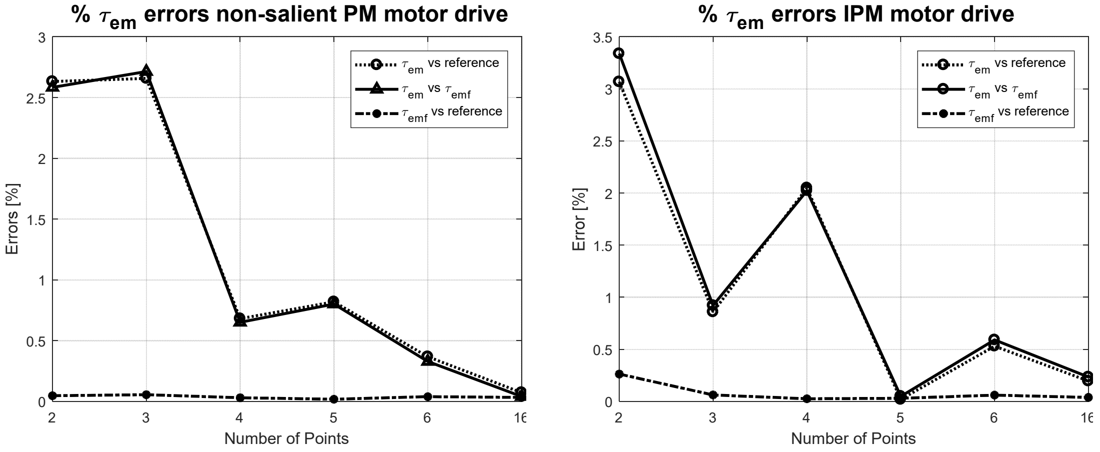
The selection of the step number may depend on design factors of each motor model analyzed, therefore, it cannot be generalized. The designer should analyzed and select the step number and length of the waveform to be studied for each motor case. Nevertheless, this subsection presents a numerical approach example on the selection of the step number for the algorithms. The algorithms are flexible and allow any number of steps to be entered as well as the length of the waveform to be analyzed, however, the fewer steps entered the faster its convergence. In this case, the selection of the step number is done based on a simple error analysis of the characteristic variables to check the amplitude of the errors we can make compared to a reference value by simulating with a reduced number of points. The error study was carried out for the three different brushless motor models.

This experiment consists of simulating a fixed operation point with two, three, four, five, six and sixteen step number points in one-sixth of an electrical period for mean values evaluation and compare the characteristic variable results with a reference value that is considered with adequate resolution. This reference value, in this case is selected to be 100 points in one electrical period since with this we assure that we can reach up to the 50th harmonic having at least two points per harmonic and for a mean value computation we considered it enough.

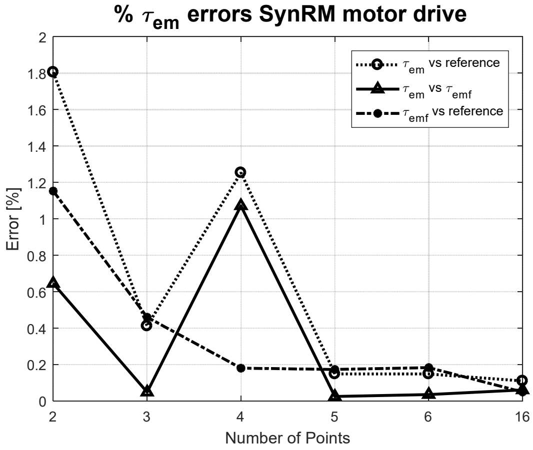
The motors are referred as the non-salient, the IPM and the SynRM motor. Fig. A-3 shows the torque errors for the three motors for a fixed current point. $\tau_{en}$is the obtained electromagnetic torque mean value directly from FEM results i.e., mean value of the output torque waveform for each specified step number. τ_{emf} is the torque computed with (3).

$$\tau_{emf} = \frac{3}{2} p\left(\psi_{ds} I_{qs} - \psi_{qs} I_{ds}\right) \quad (3)$$

It was found that the highest error is made in τ_{em} for a few number of points of simulation, which is expected since depending the torque ripple of each motor the mean value can be estimated wrong if the selection of the points are not carefully chosen. On the other hand, the error of τ_{emf} is smaller. It approximates the reference better due to the flux linkage approximation of the waveform in one-sixth of the electrical period is better compared to the torque waveform approximation. Therefore, the reason to use the equations with dq fluxes for characteristics computation.



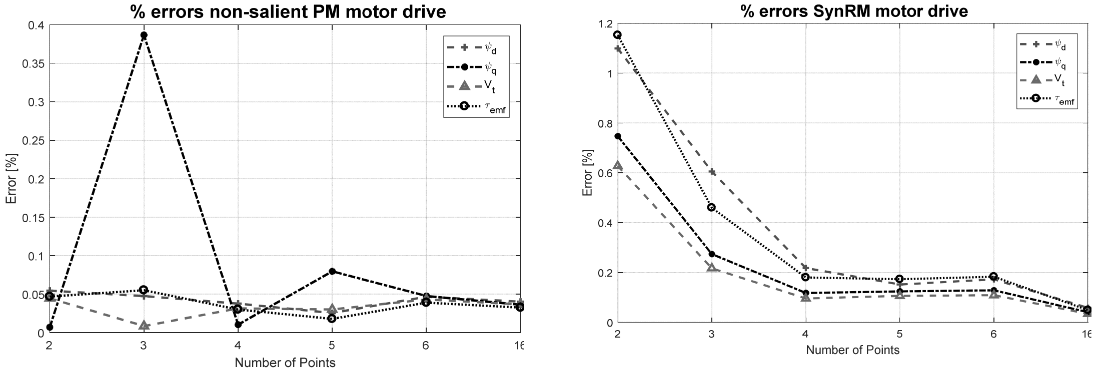
(**a**) 18 ∠ 30° (**b**) 15 ∠ 60°



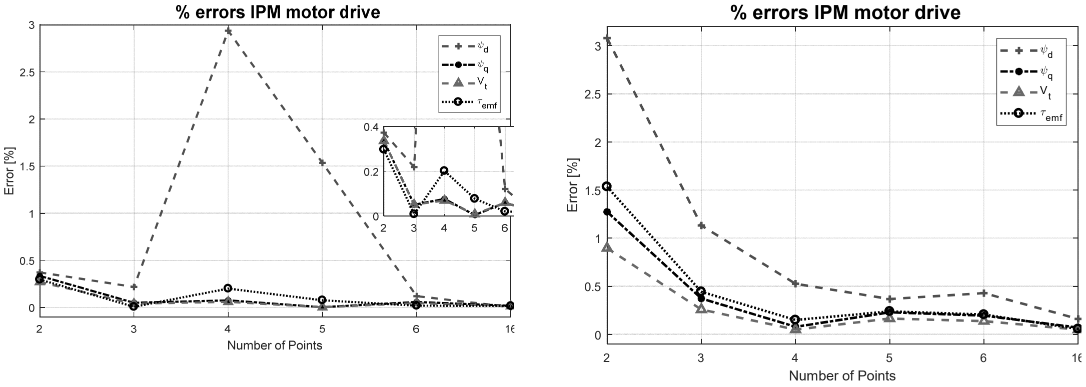
(**c**) 10 ∠ 55°

Fig. A-3. Torque errors for each motor on a fixed current phasor.

The errors for the characteristic variables of the three motor models are shown in Fig. A-4 The errors for two different currents are illustrated in the same figure for the IPM motor with the purpose of showing that the error amplitudes are similar for different points of operation. Based on the results, knowing that for few number of points in one-sixth of an electric period the error found was below 5% for the three motors analyzed, as example, a step number of 2 for the non-salient motor, and a step number of 3 for the rest is selected.



(**a**) 18 ∠ 30° (**b**) 10 ∠ 55°



(**c**) 15 ∠ 60° (**d**) 18 ∠ 70°

Fig. A-4. Errors of characteristics variables for each motor on a fixed current phasor.

Brushless Motors Speed Characteristics Computation:

To find the speed characteristics the algorithms are applied like if having no or few knowledge of the motor behavior. For example, we know that for a non-salient motor the MTPA curve is on the q axis ($I_d = 0$). This could be computed faster with Algorithm 1 or Algorithm 2 depending the limit instead of applying Algorithm 8 another example is that I_{ch} for the SynRM is zero and no algorithm is needed to find it so no assumption is made and all points are found. This is done to show the time of execution of each algorithm in a worst-case scenario.

Table A-2. Inputs used in the algorithms for the brushless motors.

Algorithm	Variable	Description	non-salient	IPM	SynRM
3	$\varepsilon_{\psi_s^d}$	Flux d tolerance	1×10^{-7}	1×10^{-7}	1×10^{-7}
	I_{S_0}	Current initial guess	5	10	5
8	$\varepsilon_{V_{ref}}$	Voltage tolerance	1×10^{-2}	1×10^{-2}	1×10^{-2}
	I_{s_0}	Current initial guess	10	20	10
	I_{s_1}	Current second guess	12	5	5
	$\bar{\gamma}$	Gamma angle array	[0, …, 75] in 10 steps	[0, …, 75] in 10 steps	[0, …, 75] in 10 steps
9	$\bar{\gamma}$	Gamma angle array	[0, …, 89] in 10 steps	[0, …, 89] in 10 steps	[0, …, 89] in 10 steps
	$\overrightarrow{I_{s\_v}}$	Current array	[20.34, ..., 19.55] in 6 steps	[13.33, ..., 10.77] in 8 steps	[14.15, ..., 5.17] in 8 steps
6	$\varepsilon_{V_{ref}}$	Voltage tolerance	1×10^{-2}	1×10^{-2}	1×10^{-2}
	N_0	Speed initial guess	166	166	166
	$\bar{\gamma}$	Gamma angle array	[0.0, …, 64.76] in 5 steps	[58.3, …, 83.61] in 5 steps	[62.9, …, 80.17] in 5 steps
5	$I_{S_{ref}}$	Current limit	6.78	4.43	4.71
	$\bar{\gamma}$	Gamma angle array	[0, …, 75] in 10 steps	[30, …, 65] in 5 steps	[30, …, 65] in 5 steps
7	N_{ref}	Speed reference	550	570	350
	$\varepsilon_{V_{ref}}$	Voltage tolerance	1×10^{-2}	1×10^{-2}	1×10^{-2}
	$\overline{I_d}$	Current d axis array	[18.62, …, 28] in 9 steps	[10.26, …, 23] in 12 steps	[5, …, 20] in 15 steps

Table A-2 shows the inputs applied to each algorithm listed in the order to compute the speed characteristics for the brushless machines. As aforementioned in Section 5.1 we are using 2 step number for the non-salient motor and 3 for the rest.

The time of execution in seconds as well as the number of iterations of convergence of each algorithm with the inputs aforementioned are presented in Table A-3. The large number of iterations for Algorithm 9 are due to Algorithm 6 which applies the Newton's method therefore most of the iterations are computed outside FEM. It can be noticed the time of execution varies depending the algorithm having the worst case for Algorithm 9 that takes 30 min to output the results. In general, the time of executions of each algorithm can be reduced by applying a better design criteria to the inputs, this is further explain in the next subsections.

Table A-3. Times of execution and number of iterations.

Parameter	Alg.	Non-Salient		IPM		SynRM	
		Time [s]	**No. Iterations**	**Time [s]**	**No. Iterations**	**Time [s]**	**No. Iterations**
I_{ch}	3	178.02	7	75.65	6	176.24	6
Mode 1 limit	8	358.39	9	1935.58	27	1116.20	27
Mode 2 limit	9	1790.15	88	1946.89	83	1790.15	87
FW point	6	174.69	17	112.90	16	112.32	15
MTPA point	5	135.31	4	115.38	6	121.04	6
MTPV point	7	595.28	14	646.88	20	562.298	23

The circle diagram and speed characteristic curves of each motor are now shown.

Non-Salient PM Drive

The non-salient PM is a 10 pole pair ferrite-magnet motor. The process to obtain the speed characteristic is the following. First, using Algorithm 3 the I_{ch} is computed. Then, mode 1 limit is computed with Algorithm 8. Mode 2 limit, CPSR and MPSR points are computed using Algorithm 9, after this with Algorithm 6 two MTPA points below mode 1 limit are obtained. Finally, three speeds greater than mode 2 limit speed MTPV points are obtained with Algorithm 7.

Table A-4 shows a comparison of the three different limits versus the reference value having the highest error of 2.1% in the mode 2 limit. Based on the authors' experience the errors and time of execution are acceptable for practical purposes. Time can be reduced by applying the algorithms with better design criteria, e.g., for Algorithm 9, which has the longest time, depending the wanted precision the number of currents to be swept can be reduced as well as narrowing the angle vector (initial and end value) to be swept. All the results are plotted in Fig. A-5, the limits of the different modes are marked with a dash line. In the circle diagram the MPSR is marked with an 'X'. For this motor the CPSR is infinite and the MPSR is 2.91 found at the boundary point of FW and MTPV.

Table A-4. Results comparison with reference value for the non-salient motor drive.

Parameter	Alg.	Alg. Result	Reference Value	Error
I_{ch} [A]	3	18.62	18.60	0.11%
Mode 1 limit [A]	8	20.34 ∠ 0°	20.59 ∠ 0°	1.2%
Mode 2 limit [A]	9	20.34 ∠ 64.76°	20.59 ∠ 64.60	2.1%

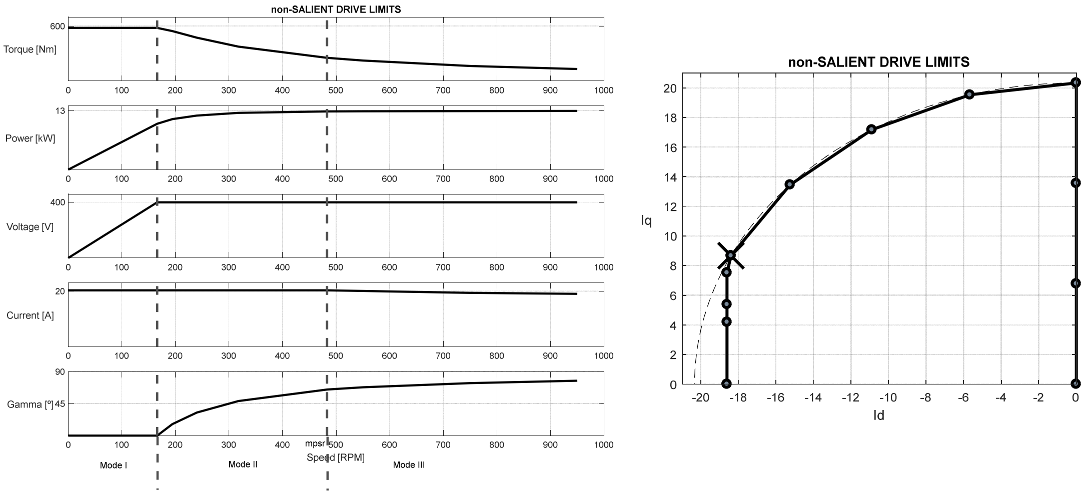


Fig. A-5. Speed characteristics results for the non-salient motor drive.

IPM Drive

The IPM is the same SynRM used in this paper with ferrite magnets in the slots of the rotor. The process carried out is the same as in for the non-salient motor with four more points computed on the MTPV curve. Table A-5 shows the comparison of the limits with the reference value. All computed points are shown in Fig. A-6. On the circle diagram the MPSR and the CPSR points are marked with an 'X'. The MPSR of this motor is 1.51 found in the mode 2 region and the CPSR is 3.2 found on the MTPV curve or in the mode 3 region, which is the expected behavior for an IPM.

Table A-5. Results comparison with reference value for the IPM drive.

Parameter	Alg.	Alg. Result	Reference Value	Error
I_{ch} [A]	3	10.26	10.15	1.08%
Mode 1 limit [A]	8	13.33 ∠ 58.30°	13.30 ∠ 58.30°	0.23%
Mode 2 limit [A]	9	13.33 ∠ 83.79°	13.30 ∠ 83.80º	0.23%

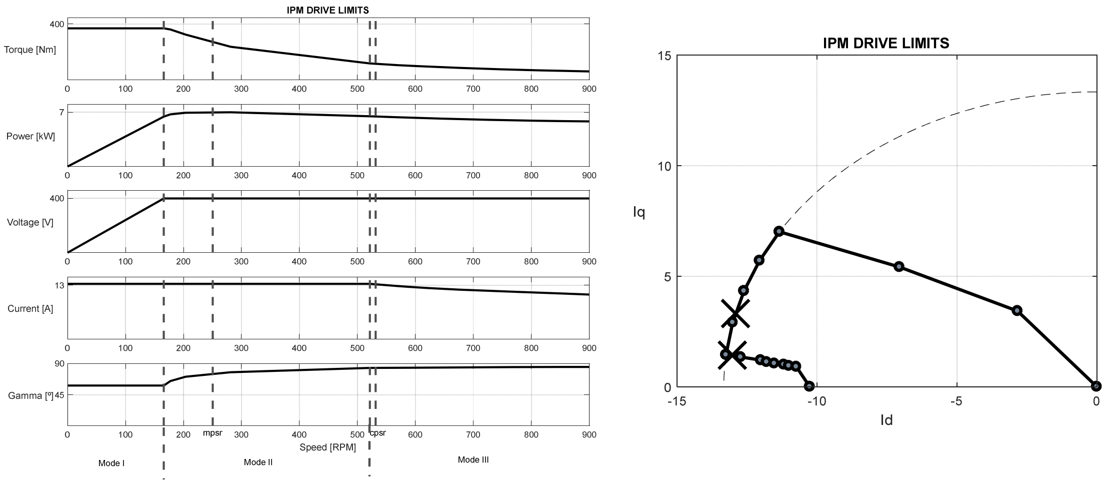


Fig. A-6. Speed characteristics results for the IPM drive.

SynRM Drive

The SynRM is a three pole pair motor. The process followed is the same as in the non-salient motor with one more points computed on the MTPV curve. Table A-6 shows the comparison of the limits with the reference value. All computed points are shown in Fig. A-7. The MPSR of this motor is 1.11 and the CPSR is 1.3 both found in the mode 2 region. On the circle diagram the MPSR and the CPSR points are marked with an ‘X’.

Table A-6. Results comparison with reference value for the SynRM drive.

Parameter	Alg.	Alg. Result	Reference value	Error
I_{ch} [A]	3	-2.63×10^{-18}	0.0	0%
Mode 1 limit [A]	8	14.15 ∠ 62.90°	14.19 ∠ 62.8°	0.28%
Mode 2 limit [A]	9	14.15 ∠ 80.17°	14.19 ∠ 80.17°	0.28%

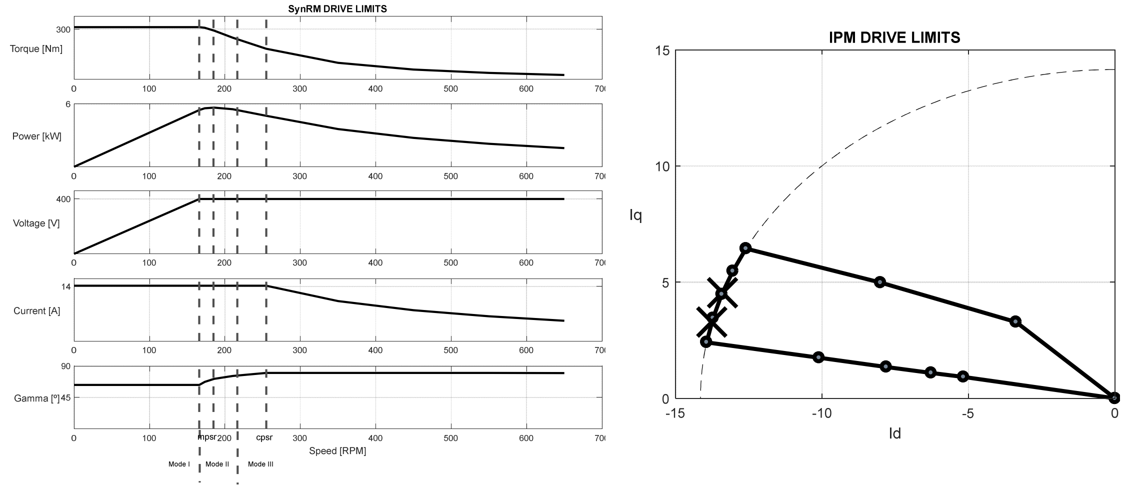


Fig. A-7. Speed characteristics results for the SynRM drive.

Appendix C **KBeMotorProfile stereotypes**

This section shows the list of stereotypes to develop the knowledge-base eMotor profile (KBeMotorProfile) for the proposed KBES framework. The stereotypes are classified per view in the following tables.

Table A-7. Structure view stereotypes

Stereotype	Applied to
Structure	Class: Block
Product	Class: Block
Assembly	Class: Block
part	Class: Block
Composite Feature	Class: Block
feature	Class: Block
has part	Association end
is part of	Association end
has composite feature	Association end
is composite feature of	Association end
has feature	Association end

Table A-8. Function view stereotypes

Stereotype	Applied to
Function	Class:Block
Principle of solution	Class: Block
has function	Association end
is a function of	Association end
Technical solutions	Class: Block
realized by	Association class
is a principle solution of	Association end
has principle of solution	Association end
has technical solution	Association end
is a technical solution of	Association end
has constraint	Association end
is constraint of	Association end
embodied by	Association class
linked to	Association end
Optional	Association class

Table A-9. Technology view stereotypes.

Stereotype	Applied to
Technology	Class: Block
Manufacturing process	Class: Block
Material	Class: Block
has material	Association end
is material of	Association end
has manufacturing process	Association end
is manufacturing process of	Association end
Test	Class: Block
has technology	Association end
is technology of	Association end
has test	Association end
is test of	Association end

Table A-10. Representation view stereotypes.

Stereotype	Applied to
Representation	Class:Block
CAE	Class:Block
CAD	Class:Block
Analytical	Class:Block
CAE property	Class:Block
CAD property	Class:Block
Analytical property	Class:Block
Has representation	Association end
Has property	Association end
Is property of	Association end
Is representation of	Association end

Table A-11. Micro-process view stereotypes

Stereotype	Applied to
Approach Act	Activity diagram
Approach CB	Class:Constraint block
Criteria	Comment
Cross-domain objective	Class:Block
Defined by	Dependency
Domain	Class:Block
Has approach	Association end
Has expanded indication	Association end
Has objective	Association end
Indication Act	Activity diagram
Indication action	State
Indication CB	Class:Constraint block
Indication cnst	Constraint
Is approach of	Association end
Is expanded indication of	Association end
Is objective of	Association end
Objective	Class:Block
Validated by	Dependency

Appendix D **Micro-level KBE application: ModeladoScript**

ModeladoScript is composed of three main function:

1. Motorcad_calc function
2. ModeladoScript function
3. export2xml function

ModeladoScript is a micro-level KBE application contained by the macro-level KBE application presented in Chapter 6. This application was an initial implementation that helped to identify the needs and clarify processes in order to achieve the proposed KBE system framework. The reason why the process (or other knowledge) is not represented following the knowledge representation presented in the thesis (using the KBeMotorProfile) is to show the reader how the proposed knowledge representation leverage the structure for information retrieval. Here the knowledge is presented in text and simple flowcharts that makes hard the retrieval of information from a knowledge base.

Function name: Motorcadcalc

This section describes the architecture and algorithm of the function called motorcadcalc. A detailed description of inputs/outputs, parameters and algorithms is presented.

Scope

Motorcadcalc is a function intended to ease the workload of the electrical machine designer by simulating in Motorcad different power losses (iron and copper losses) either for a given specific value or a given range of values, returning as outputs the rise in copper and magnet temperature.

Function context

The function is to be developed in vbscript language, the function can be called from any script and from any other language i.e. matlab, python, etc. The rise of copper temperature and magnet temperatures are the outputs of the function and the values are to be exported to excel. The manual process followed by the designer is now described.

1. Insert Pcu and Ph losses values in Motorcad.
2. Execute simulation in Motorcad.
3. Export results to excel file.
4. Copy and paste results to a predetermined excel file.
5. Subtract the ambient temperature to obtain the temperature rise.
6. Repeat process for next values.

Because the user already has scripts that uses the values from the excel files previously created, the way the results are exported must followed the same format which was provided by the domain expert.

User characteristics

The user of the function must have knowledge in electromagnetics design software, therefore it is intended for electrical machines designers with some experience in Motorcad software.

Scripts and Functions

The whole motorcadcalc function consists of three parts which are:

1. **Main.vbs**, main script where input parameters are entered, it also contains a subroutine to read other vbs files to execute the code within. The function calls are motorcad_calc.vbs and export_xls.vbs.
2. **Motorcad_calc.vbs**, this is the function where the principal code is written, it is executed given the input parameters and returns the output parameters aforementioned.
3. **xls_export.vbs,** this function is only created to export the output values to an excel file following the format previously requested by the designer (see motorcad_calc requirements document).

Function Activity diagram

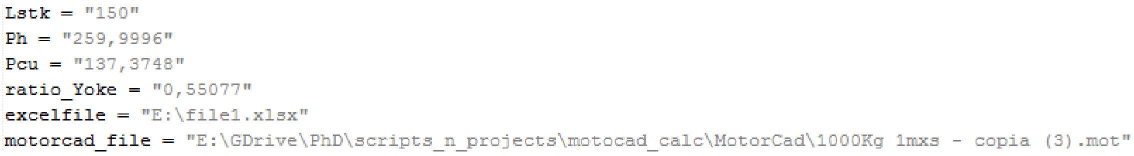
The function activity diagram is presented in Fig. A-8.

Algorithm

The algorithm is illustrated in Fig. A-9.

User interface

The user interface will only be the main script by introducing the inputs to be simulated in motorcad. Hence the input data is entered as follows:



```
Lstk = "150"
Ph = "259,9996"
Pcu = "137,3748"
ratio_Yoke = "0,55077"
excelfile = "E:\file1.xlsx"
motorcad_file = "E:\GDrive\PhD\scripts_n_projects\motocad_calc\MotorCad\1000Kg 1mxs - copia (3).mot"
```

The outputs values are saved in an xls file with the following format:

1. One worksheet per Lstk simulated.
2. dTcu starts saving from row 6, column 19.
3. dTm starts saving from row 23, column 19.

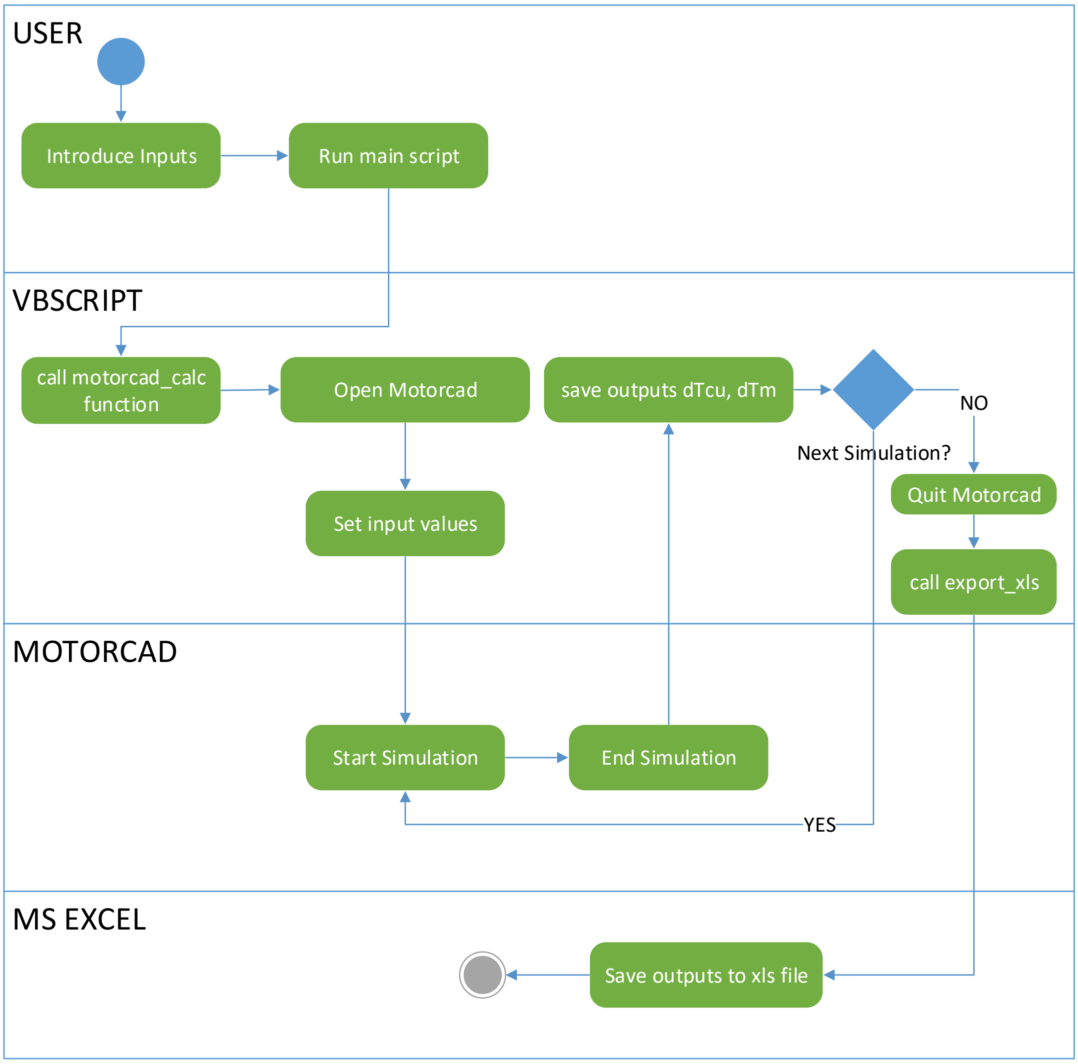


Fig. A-8. Motorcadcalc architecture and activity diagram

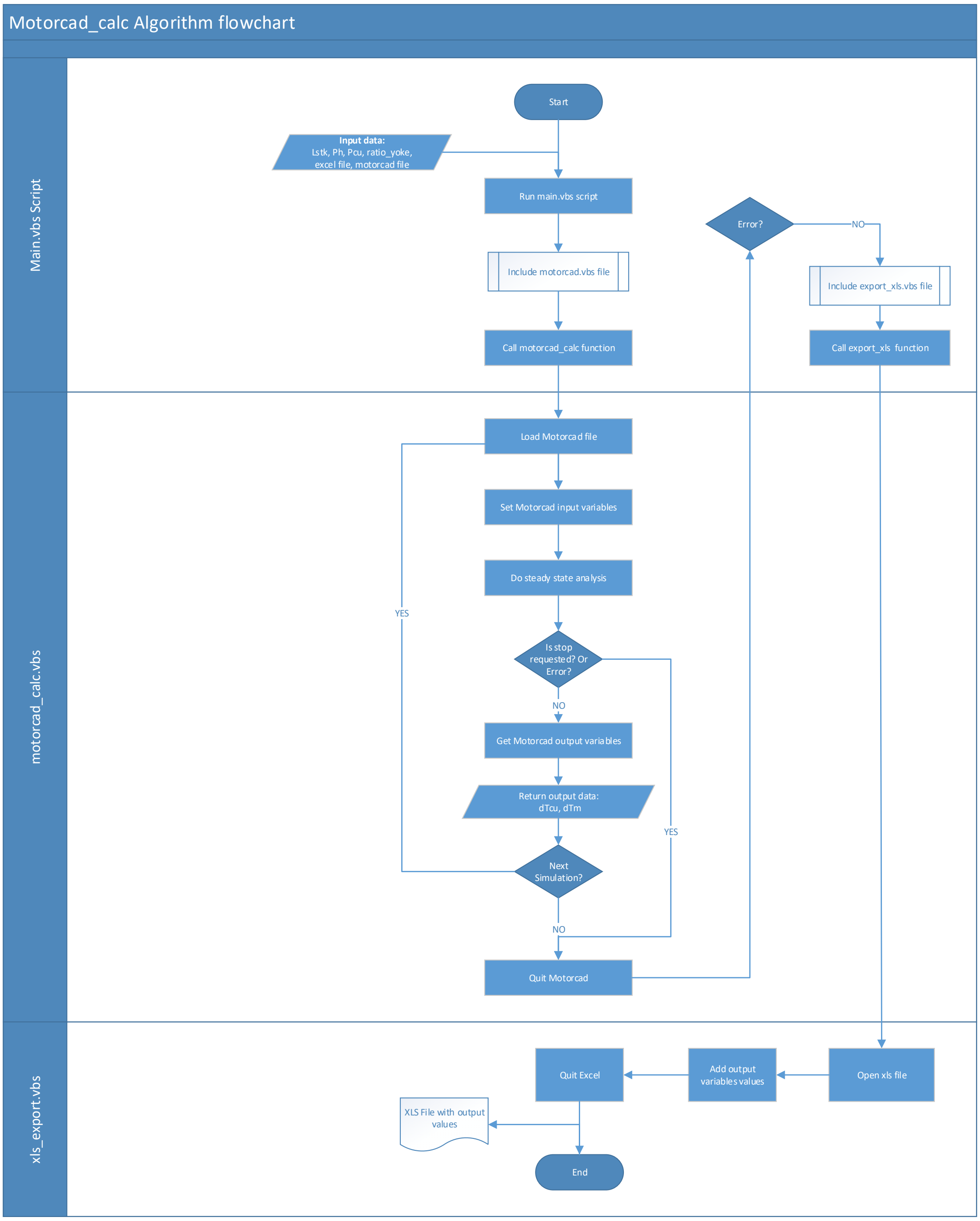


Fig. A-9. Motorcadcalc algorithm.

Function name: ModeladoScript

Scope

ModeladoScript is a tool for dimensioning motors. Function created with the intention to ease the workload of the electrical machine designer by extrapolating Nph and Lstk from a reference motor simulated in a FEM tool and evaluating different torque requirements. Thermal analysis is also taken into account by integrating motorcadcalc function which simulates different power losses (iron and copper losses) in Motorcad returning as outputs the temperature rise in copper and magnet to evaluate the motor in RMS torque load condition.

Function context

The function is to be developed in Matlab language. The function supports the manual process followed by the designer to analyze different operating points of the motor to see if it fulfills requirements. The process is described below. With the help of ModeladoScript the designer can obtain the operating points faster.

1. Stablish the length of the motor.
2. Adjust the number of turns for the required load at ambient temperature.
3. Validate the maximum torque at cold condition by calculating the maximum point value and make sure the maximum voltage is not exceeded. If the maximum torque is not fulfil, go back to step 2 and reduce the number of turns avoiding the rated conditions does not surpass the highest limit. If reducing the number of turns is not possible, go to step 1 and increase the motor length.
4. Calculate the motor losses at rms torque. Previously the current corresponding the rms torque must be adjusted. For instance, if the rms torque is 70% of the rated torque, current would not be 70% of rated value rather it is generally higher due to the not linear performance. Thus, iterations should be carried out from the 70% current value.
5. Once the losses are obtained, calculate the copper and magnet temperatures at RMS torque condition. If the motor surpass specified values go to step 1 and increase the motor length.
6. Validate the maximum torque in hot conditions by calculating the torque and proving the highest limit in voltage is not surpass. If the motor cannot reach the maximum torque go to step 2 and reduce the number of turns if this is not possible go to step 1 and change the length of the machine.
7. Calculate rated values in hot condition.

In addition, the function also includes features that were not taken into account. Such features are:

1. Resistance and leakage inductance calculation. This was done separately in an excel worksheet.
2. Thermal analysis was not included.
3. No torque or voltage inputs could be entered for interpolation only currents was the tool capable to process as inputs.
4. Data was not presented in an easy readable format.

User characteristics

The user of the function must have knowledge in electromagnetics design software, therefore it is intended for electrical machines designers with experience in Cedrat Flux, Matlab and Motorcad software.ModeladoScript design

Function activity diagram

The function activity diagram is presented in Fig. A-13.

Algorithm

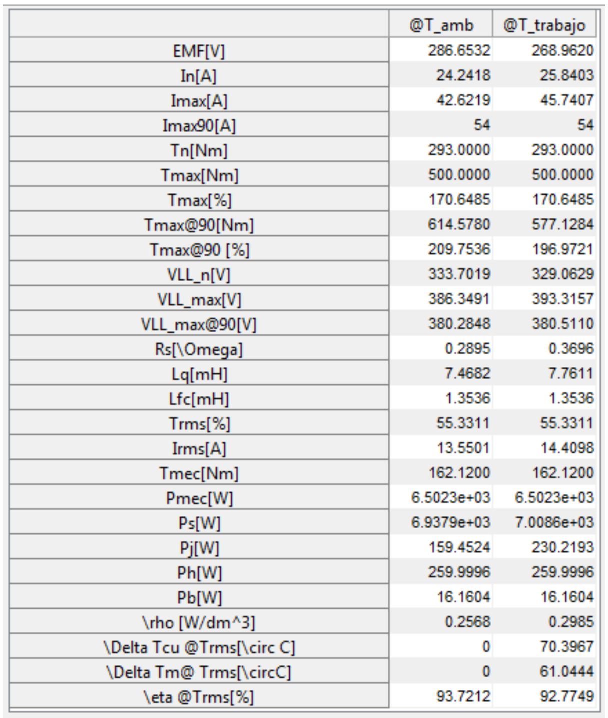
The algorithm is illustrated in Fig. A-14.

User interface

The user interface is the main script ModeladoScript.m where the inputs are entered and the uitable where the output values are presented. Fig. A-10 shows the uitable with the output variables values regarding motor performance for a particular dimension and specification. Besides the table the script also creates an xls file with the same values with the name tabla_dimensionado where the values are replace for each execution of the script.

FEM Results xls file Format

To read and import the FEM simulation results with ModeladoScript the file format must be as described in Fig. A-11.



	@T_amb	@T_trabajo
EMF[V]	286.6532	268.9620
In[A]	24.2418	25.8403
Imax[A]	42.6219	45.7407
Imax90[A]	54	54
Tn[Nm]	293.0000	293.0000
Tmax[Nm]	500.0000	500.0000
Tmax[%]	170.6485	170.6485
Tmax@90[Nm]	614.5780	577.1284
Tmax@90 [%]	209.7536	196.9721
VLL_n[V]	333.7019	329.0629
VLL_max[V]	386.3491	393.3157
VLL_max@90[V]	380.2848	380.5110
Rs[\Omega]	0.2895	0.3696
Lq[mH]	7.4682	7.7611
Lfc[mH]	1.3536	1.3536
Trms[%]	55.3311	55.3311
Irms[A]	13.5501	14.4098
Tmec[Nm]	162.1200	162.1200
Pmec[W]	6.5023e+03	6.5023e+03
Ps[W]	6.9379e+03	7.0086e+03
Pj[W]	159.4524	230.2193
Ph[W]	259.9996	259.9996
Pb[W]	16.1604	16.1604
\rho [W/dm^3]	0.2568	0.2985
\Delta Tcu @Trms[\circ C]	0	70.3967
\Delta Tm@ Trms[\circC]	0	61.0444
\eta @Trms[%]	93.7212	92.7749

Fig. A-10. Uitable with output variables.

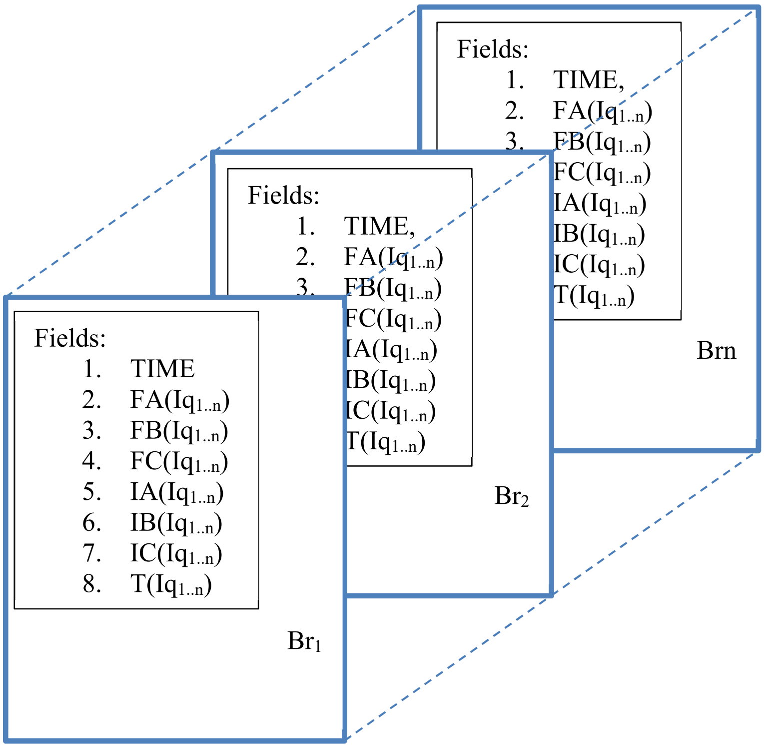
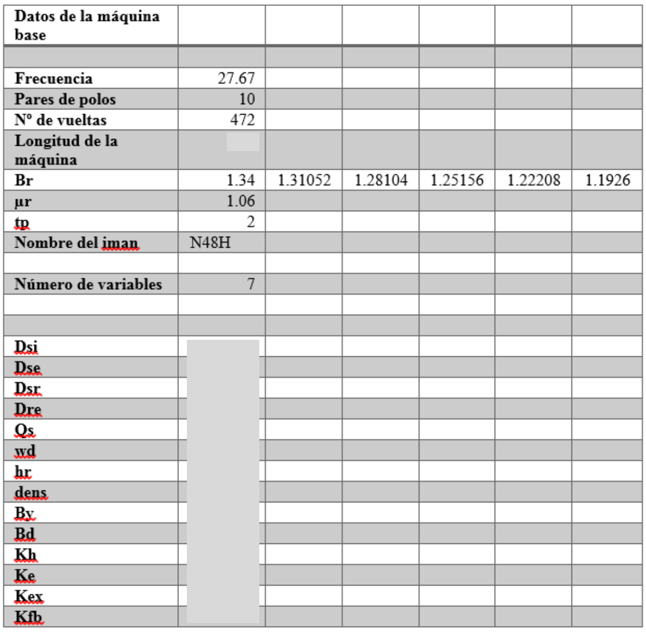


Fig. A-11. File format for FEM simulation results.

It is mandatory to have the last worksheet named "informacion" with the file format and data shown in Fig. A-12.



Datos de la máquina base						
Frecuencia	27.67					
Pares de polos	10					
Nº de vueltas	472					
Longitud de la máquina	[redacted]					
Br	1.34	1.31052	1.28104	1.25156	1.22208	1.1926
µr	1.06					
tp	2					
Nombre del iman	N48H					
Número de variables	7					
Dsi	[redacted]					
Dse	[redacted]					
Dsr	[redacted]					
Dre	[redacted]					
Qs	[redacted]					
wd	[redacted]					
hr	[redacted]					
dens	[redacted]					
By	[redacted]					
Bd	[redacted]					
Kh	[redacted]					
Ke	[redacted]					
Kex	[redacted]					
Kfb	[redacted]					

Fig. A-12. Screenshot of worksheet "informacion" data.

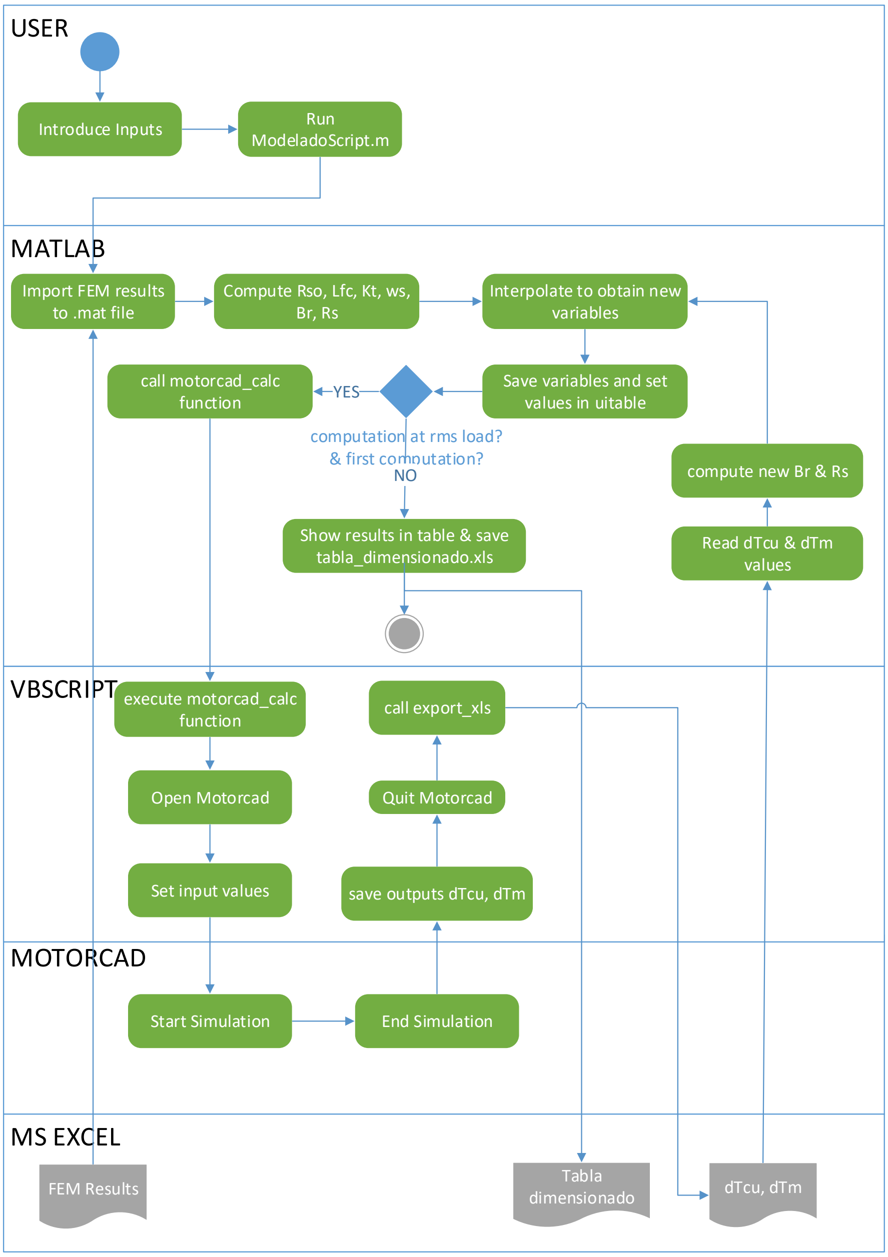


Fig. A-13. ModeladoScript activity diagram.

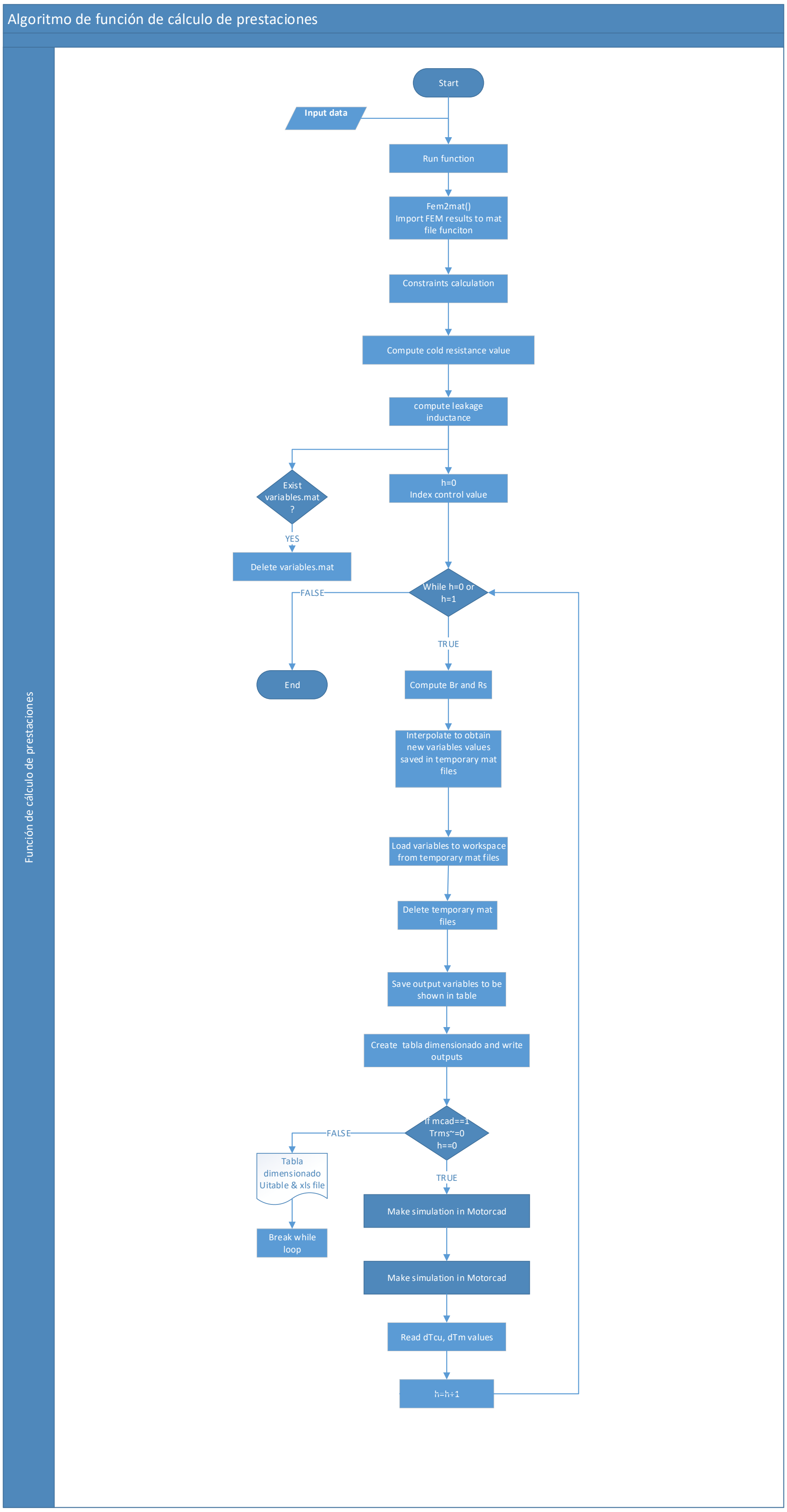
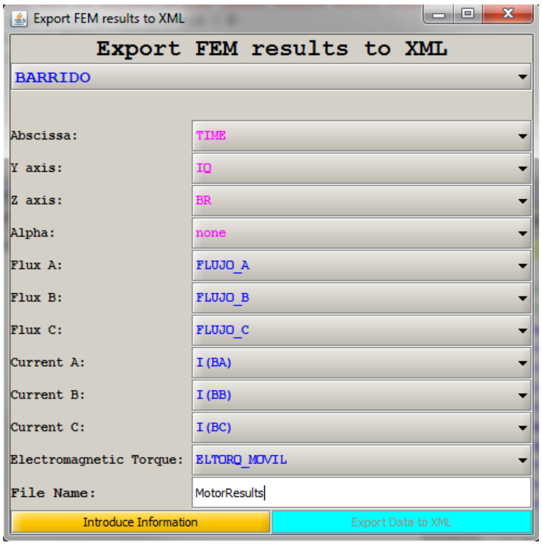


Fig. A-14. ModeladoScript algorithm flowchart.

Function name: Export2XML

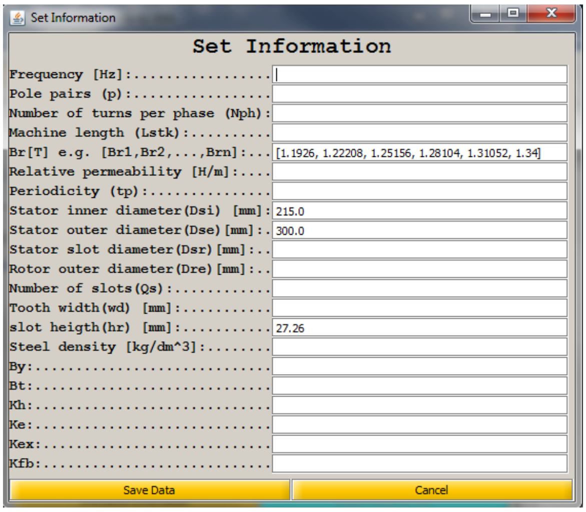
This application was developed to export the simulation results in XML format from the FEM tool. The data structure is the same as described in ModeladoScript but instead of making it in Microsoft Excel files the devised tool exports the files in XML format without additional effort from the end-user. ModeladoScript was modified in order to accept this type of format, however, the user still has the capability of importing XLSX files. This tool overcomes the problem of variable identification, every designer put different names to the variables making hard to export the results. Therefore, the tool standardize the process by making the user select from each combobox the variable for the requested parameter. Fig. A-15 shows the user interface of the main window.



	BARRIDO
Abscissa:	TIME
Y axis:	IQ
Z axis:	BR
Alpha:	none
Flux A:	FLUJO_A
Flux B:	FLUJO_B
Flux C:	FLUJO_C
Current A:	I(BA)
Current B:	I(BB)
Current C:	I(BC)
Electromagnetic Torque:	ELTORQ_MOVIL
File Name:	MotorResults

Fig. A-15. Application main window.

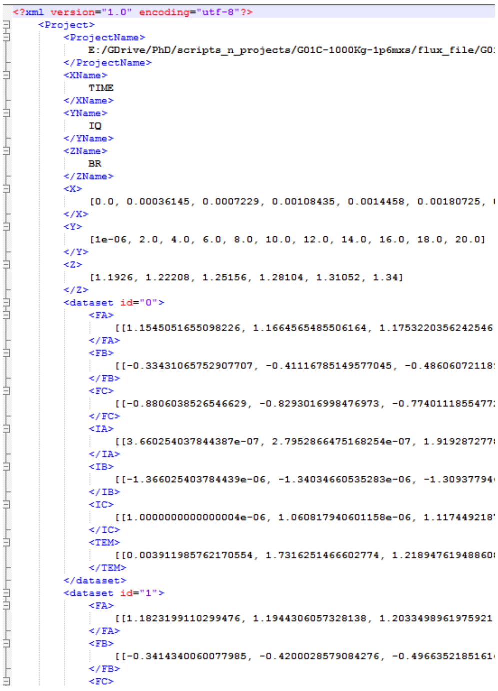
The user still has to insert the information requested of the motor in order to postprocess the results in ModeladoScript, see Fig. A-15.



Frequency [Hz]:	
Pole pairs (p):	
Number of turns per phase (Nph):	
Machine length (Lstk):	
Br[T] e.g. [Br1,Br2,...,Brn]:	[1.1926, 1.22208, 1.25156, 1.28104, 1.31052, 1.34]
Relative permeability [H/m]:	
Periodicity (tp):	
Stator inner diameter(Dsi) [mm]:	215.0
Stator outer diameter(Dse)[mm]:	300.0
Stator slot diameter(Dsr)[mm]:	
Rotor outer diameter(Dre)[mm]:	
Number of slots(Qs):	
Tooth width(wd) [mm]:	
slot heigth(hr) [mm]:	27.26
Steel density [kg/dm^3]:	
By:	
Bt:	
Kh:	
Ke:	
Kex:	
Kfb:	

Fig. A-16. Insert information window.

After this, the user can press the export data to xml button and the results are export into XML structure as shown in Fig. A-17. The application has also been extended to admit a fourth dimension and the xml has been re-structure but still in developing process is the modification of ModeladoScript which now only accepts the current module as inputs. To date this is still an undergoing improvement on the tools.



```
<?xml version="1.0" encoding="utf-8"?>
    <Project>
        <ProjectName>
            E:/GDrive/PhD/scripts_n_projects/G01C-1000Kg-1p6mxs/flux_file/G0
        </ProjectName>
        <XName>
            TIME
        </XName>
        <YName>
            IQ
        </YName>
        <ZName>
            BR
        </ZName>
        <X>
            [0.0, 0.00036145, 0.0007229, 0.00108435, 0.0014458, 0.00180725,
        </X>
        <Y>
            [1e-06, 2.0, 4.0, 6.0, 8.0, 10.0, 12.0, 14.0, 16.0, 18.0, 20.0]
        </Y>
        <Z>
            [1.1926, 1.22208, 1.25156, 1.28104, 1.31052, 1.34]
        </Z>
        <dataset id="0">
            <FA>
                [[1.1545051655098226, 1.1664565485506164, 1.1753220356242546
            </FA>
            <FB>
                [[-0.33431065752907707, -0.41116785149577045, -0.48606072118
            </FB>
            <FC>
                [[-0.8806038526546629, -0.8293016998476973, -0.774011185547
            </FC>
            <IA>
                [[3.660254037844387e-07, 2.7952866475168254e-07, 1.91928727
            </IA>
            <IB>
                [[-1.366025403784439e-06, -1.34034660535283e-06, -1.30937794
            </IB>
            <IC>
                [[1.0000000000000004e-06, 1.060817940601158e-06, 1.117449218
            </IC>
            <TEM>
                [[0.003911985762170554, 1.7316251466602774, 1.21894761948860
            </TEM>
        </dataset>
        <dataset id="1">
            <FA>
                [[1.1823199110299476, 1.1944306057328138, 1.2033498961975921
            </FA>
            <FB>
                [[-0.3414340060077985, -0.4200028579084276, -0.4966352185161
            </FB>
            <FC>
```

Fig. A-17. XML structure, file example.