Universidade do Minho
Escola de Engenharia
Departamento de Engenharia Mecânica

Automatic and intelligent integration of manufacture standardized specifications to support Product Life Cycle

An ontology based methodology

PhD thesis by

João Pedro Mendonça de Assunção da Silva

Supervised by: Professor Ricardo Jardim Gonçalves

and Professor António Caetano Monteiro

Guimarães
September, 2009

To my lovely wife Paula,

and our daughters Ana João e Sofia

Vós, Portugueses, poucos quanto fortes,
Que o fraco poder vosso não pesais;
Vós, que à custa de vossas várias mortes
A lei da vida eterna dilatais:
Assi do Céu deitadas são as sortes
Que vós, por muito poucos que sejais,
Muito façais na santa Cristandade.
Que tanto, ó Cristo, exaltas a humildade!

Camões, Lusiadas, Canto VII, verso 3

Acknowledgements

This thesis would not be possible without the valuable support and help provided by several individuals, institutions, industrial companies and friends which I had the opportunity to work closely. Here I refer those whose contribution was of major importance. I want to express my gratitude to:

Professor António Caetano Monteiro, supervisor of this research work, for his scientific guidance, valuable suggestions, encouragement and valuable revision of the text.

Professor Ricardo Jardim Gonçalves, my research and scientific supervisor, for the continuous encouragement and availability even in difficult periods of his duties. I hope to be and continue meritorious of his friendship and close cooperation.

Professor Adolfo Steiger-Garção, mentor of this research work, for his scientific guidance and valuable suggestions.

The Departamento de Engenharia Mecânica of the Universidade do Minho for the facilities and equipment provided and for supporting this research program.

The UNINOVA of the Universidade Nova de Lisboa, where part of this work was made, for the facilities and equipment provided, the kind hospitality and liaison with several European projects;

The Fundação Calouste Gulbenkian and Fundação para a Ciência e Tecnologia for the financial support of the working meetings in USA and in Europe.

Finally, the conclusion of this thesis would not be a reality without the valuable help of special persons in my life. I would like to specially recognize some of them here.

I would like to thank to all talented colleagues of Departamento de Engenharia Mecânica, for their friendship, support and clarity moments. A special thank to José Luís e José Carlos Teixeira.

I appreciate all the colleagues in my research group for sharing precious ideas, in particular Professor Santos Pais, for the richness of our long talks and inspiring discussions;

I am very grateful for the fruitful discussions and all the fun I had with colleagues of UNINOVA, while contributing to several European research projects, such as ATHENA, INTEROP NoE and funSTEP, who created the most friendly, memorable, and stimulating working atmosphere. I have benefited from their efforts far beyond this work.

In particular, I would like to thank to João Sarraipa for his availability, ideas and support during the framework development phase, which contribute was indispensable for the success of the envisaged solution.

A special acknowledge to Eng Sofia Silva from PLASFIL, Eng Nuno Damas from MicroPlásticos, Eng. Ramiro Pereira from INEMER, Eng Nuno Faria from ETMA and Dr. Luís Sousa Lopes from INFOS, which industrial knowledge and insights were the basis for my inspiration during the work;

Last, but not least, I am deeply grateful to my family and friends for always being there (yes, they are my Milky Way). Thanks for your continuous care, patience and endless support.

A special thank to my wife Paula for the unflinching support and for her patience and love during all these years.

Finally thanks Ana João e Sofia, you are my inspiration thoughts and surely dad regrets all the untold stories while you get asleep.

Abstract

In the last decades, the globalization introduced significant changes in the product's lifecycle. A worldwide market advantageously offered a vast range of Products, both in terms of variety and quality. In consequence, markets progressively demand highly customized products with short life cycle. Computational resources provided an important contribute to maintain Manufacture competitiveness and a rapid adaptation to paradigm change from mass production to mass customization as well. In this environment, Enterprise and Product modeling were the best response to new requirements like flexibility, agility and intense dynamic behavior.

Enterprise Modeling enabled production convergence to an integrated virtual process. Several enterprises clearly assumed new formats like Extended Enterprises or Virtual/Agile Enterprises to guarantee product and resources coordination and management within the organization and with volatile external partners.

By the other hand, Product modeling suffered an evolution, with traditional human based resources (like technical drawings) migrating to more skilful computational product models (like CAD or CAE models). Product modeling, together with an advanced information structure, has been recognized by academic and industrial communities as the best way to integrate and co-ordinate in early Design stages the various aspects of product's lifecycle.

An early and accurate product specifications settlement is the direct consequence of the product models enrichment with additional features. Therefore, Manufacture specifications – for longtime included in technical drawings or text based notes – need to re-adapt to such reality, namely due to missing integration automation and computational support.

Recent enhancements in standard product models (like ISO 10303 STEP product data models) made a significant contribution towards product knowledge capture and information integration skills. Nevertheless, computational integration issues arise because multiple terminologies are in use along Product Life Cycle, namely due to different team backgrounds. Besides, the advent of internet claimed semantic capabilities in standard product models to a better integration with Enterprise agents.

Ontologies facilitate the computational understanding, communication and seamless interoperability between people and organizations. They allow key concepts and terms relevant in a given domain to be identified and defined in an open and unambiguous computational way. Therefore, ontologies facilitate the use and exchange of data, information, and knowledge among inter-disciplinary teams and heterogeneous systems, towards intelligent systems integration.

This work proposed a methodology to support the development of a harmonized reference ontology for a group of enterprises sharing a business domain. This methodology is based on the concept of Mediator Ontology (MO), which assists the semantic transformations between each enterprise's ontology and the referential one. The methodology makes possible each organization to keep its own terminology, glossary and ontological structures, providing seamless communication and interaction with the others.

The methodology foment the re-use of data and knowledge incorporated in the standard product models, as an effective support of collaborative engineering teams in the process of product manufacturability evaluation, anticipating validity of manufacture specifications.

Keywords: standard product data models, ISO 10303, STEP, manufacture specifications, manufacture constraints, knowledge representation, ontology building, semantic harmonization, Enterprise Modeling, Product Modeling, Ontology, Reusability and Interoperability, 3D semantic

Resumo

Nas últimas décadas, o advento da globalização introduziu mudanças significativas no ciclo de vida dos produtos. Um mercado mundial passou a oferecer vantajosamente uma gama alargada de Produtos, tanto em termos de variedade como de qualidade. Como consequência, os mercados passaram a exigir progressivamente produtos muito personalizados e com um ciclo de vida mais curto. O recurso a meios computacionais constitui um contributo importante para manter a competitividade da Manufactura e uma adaptação rápida à mudança de paradigma da Produção em massa para a personalização em massa. Neste ambiente, com muitas novas exigências tais como a flexibilidade, a agilidade e o comportamento extremamente dinâmico, a modelação das Empresas e do Produto foram a melhor solução encontrada para os meios produtivos.

A Modelação de Empresas permitiu a convergência da produção para um processo virtual integrado. Várias empresas assumiram claramente novos formatos como Empresas Estendidas ou Empresas Virtuais/Ágeis de modo a garantir coordenação e gestão do produto e de recursos, quer dentro da organização e quer com parceiros externos voláteis e/ou pontuais.

Por outro lado, a modelação de Produto sofreu uma evolução, assistindo-se à migração dos recursos tradicionais de natureza humana (como desenhos técnicos) para recurso a modelos de produtos auxiliados por computador (como CAD ou modelos CAE). A modelação de Produto, juntamente com uma estrutura de informação avançada, tem sido reconhecida pelas comunidades académicas e industriais, como a melhor forma de integrar e coordenar, na fase inicial de Design/Projecto, os multifacetados aspectos do ciclo de vida do produto.

Todas estas contribuições permitiram a estipulação de especificações dos produtos com mais antecedência e com melhor precisão. No entanto, ficaria ainda a faltar uma adaptação das especificações de Fabrico - incluídas desde sempre em desenhos técnicos ou em notas baseadas em texto – a essa nova realidade, nomeadamente pela falta de automação, de integração e de possibilidade de suporte computacional adequado.

As melhorias recentes em modelos de produto normalizados (como é exemplo o modelo de dados de produto STEP – ISO 10303) deram um contributo significativo para a inclusão de conhecimento e mecanismos de integração de informação adicional acerca do produto. Contudo, subsistiram alguns problemas de integração computacional porque várias terminologias são usadas ao longo do Ciclo de Vida do Produto, tendo em conta as diferentes vocações das equipas de Projecto e Fabrico. Por outro lado, a crescente utilização de Internet começou a necessitar de modelos de produtos com capacidades semânticas, para uma completa e profícua integração com os agentes de Modelos de Empresas Virtuais.

As ontologias facilitam o entendimento computacional entre aplicações, e como tal a melhoria da comunicação e interoperabilidade entre pessoas e organizações. As ontologias visaram que conceitos chave e termos relevantes de um determinado domínio fossem identificados e definidos de um modo computacional explicito, normalizado e inequívoco. Assim, as ontologias facilitam o uso e intercâmbio de dados, informação e conhecimento entre equipas interdisciplinares e sistemas heterogéneos, catalizando a integração de sistemas inteligentes.

Este trabalho propõe uma metodologia de apoio ao desenvolvimento de uma ontologia de referência, harmonizada para um grupo de empresas que partilhem um domínio de negócios. Esta metodologia é baseada no conceito de Ontologia Mediadora, que possibilita as transformações semânticas entre a ontologia preexistente de cada empresa e a de referência. A metodologia possibilita que cada organização mantenha a sua própria terminologia, glossário e estruturas ontológicas, proporcionando uma comunicação e interacção directa com os outros.

Esta metodologia contribui para a reutilização de dados e conhecimentos incorporados nos modelos de produto normalizados, como um apoio efectivo às equipas de engenharia no processo de avaliação da fasebilidade do produto, nomeadamente pela averiguação automática da validade das especificações de fabrico.

Palavras-chave: STEP, ISO 10303, especificações de fabrico, especificações de produto, Modelação de Empresas, Empresas Virtuais, Ontologias, harmonização de ontologias, semântica de modelos 3D

Table of Contents

Table of Figures

List of Tables

List of Acronyms

ARIS	Architecture for Information System
ATM	Asynchronous Transfer Mode
BPR	Business Process Re-Engineering
CAD	Computer Aided Design / Drawing
CAE	Computer Aided Engineering
CALS	Continuous Acquisition and Life Cycle Support
CAM	Computer Aided Manufacturing
CEN	Comité Européen de Normalisation
CIM	Computer Integrated Manufacturing
CIMOSA	Computer Integrated Manufacturing Open System Architecture
CORBA	Common Object Request Broker Architecture
DFX	Design for X (where variable X may be Manufacturing, Recycling, etc)
DIS	Draft International Standard
DL	Description Logic
DWG	Autodesk Drawing format file extension
EDI	Electronic Data Interchange
EI	Enterprise Integration
EM	Enterprising Modeling
ENV	Europäische Vornorm
ESPRIT	European Strategic Programme on Research and Development in Information Technology
EXPRESS	Standard Information modeling language ISO 10303-Part 11
FMEA	Failure Mode Effects Analysis
GERAM	Generalized Enterprise Reference Architecture and Methodology
GRAI	Graphs with Results and Activities Interrelated
HOLOS	Holonic MultiAgent Frammework (within MASSYVE)
HTML	HyperText Mark-up Language
IDEF	The family of ICAM Definition Languages, short IDEF, Integrated DEFinition
IEM	Integrated Enterprise Modeling
IFAC	International Federation of Automatic Control
IFIP	International Federation for Information Processing
IGES	Integrated
IMS	Intelligent Manufacturing Systems

IMS-CGE	Global Concurrent Engineering - EP 7752
IMS-HCS	Holonic Manufacturing System - EP 7754
ISO	International Organization for Standardization
IT	Information Technology
ICT	Information and Communication Technology
MASSYVE	MultiAgent Agile Manufacturing Scheduling Systems for Virtual Enterprises
MENTOR	Methodology for Enterprise Reference Ontology Development
MMS	Manufacturing Message System
MO	Mediator Ontology
NIIIP	National Industrial Information Infrastructure Protocols
OLE	Object Linking and Embedding
OMG	Object Management Group
OMT	Object Modeling Technique
OO	Object Oriented
OOA	Object Oriented Modeling
OPAL	Integrated Information and Process Management in Manufacturing Engineering - EP 20377
OSI	Open System Interconnect
OWL	Web Ontology Language
PERA	Purdue Enterprise Reference Architecture
PLib	ISO 13584 Standard series "Parts Library"
QCIM	Quality in CIM
QFD	Quality Function Deployment
RDF	Resource Description Framework
SET	AutoCAD LT drawing Set File
SGML	Standard Generalised Mark-up Language
SME	Small- and Medium-Size Enterprise
STEP	Standard for the Exchange of Product Model Data
TC	Technical Committee
TQM	Total Quality Management
UML	Unified Modeling Language
XML	eXtended Mark-up Language
WfMC	Workflow Management Coalition
WG	Working Group
WWW	World-Wide Web

Chapter I

I. Introduction

I.1. Motivation of the research work

For longtime, Technical Drawings populate traditional manufacturing environments. “A Drawing worth a thousand words”[1] and Technical Drawings were not exception: they were the best appropriate and universal tool to share vital information in human-centered Production Chain. More than merely carrying information, technical drawings worked as a scale product model, allowing designers and engineers to foresee much of the issues a Project always encloses.

With traditional Production paradigm relying on human labor organization, from the Manufacture point of view, such Product Model was suitable to guarantee that manufacture specifications were manually delineated and properly deployed to production agents, whatever specific function they assumed.

[1] Popular Chinese saying

Looking to the example of the orthogonal view represented in Figure 1, the drawing quickly enables a human mental image (though not complete) of the modeled product. In technical drawings, text based information is usually linked with geometrical definition of the Product, and corresponds to explicit manufacture specifications. The interpretation by humans of such kind of data will certainly determine the restrictions considered in the manufacture of the product (and vice-versa).

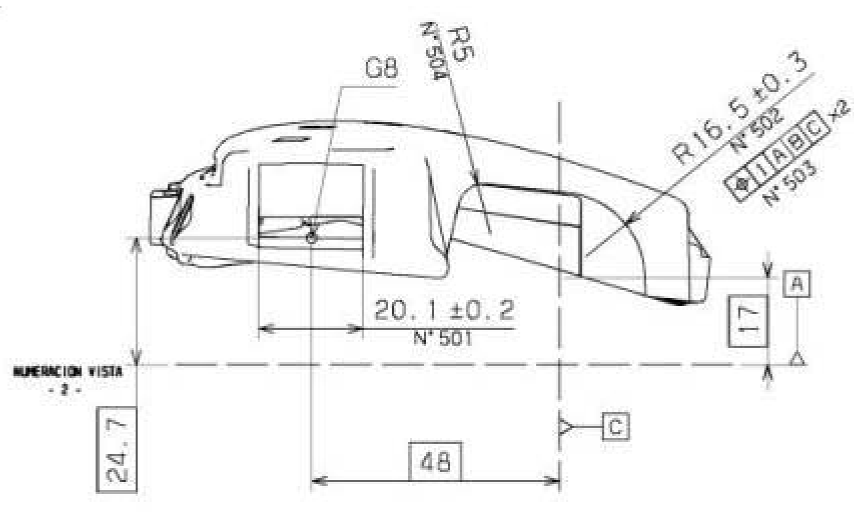


Figure 1 – Partial technical drawing with component's dimensions (geometric and topologic).

However, the introduction of computer support inevitably demanded a more suitable Product Model in order to enable aid and automation of, until then, human competences. The annotations and text based manufacture specifications became less and less compatible with computational resources meanwhile made available in the Production environment.

The international standard body ISO TC184 / SC4 developed the STEP file format, as a standard product data model. Such standardized solution was established amid proprietary solutions as a neutral information carrier, and, in manufacturing aspects, surpassed much standards, like IGES or *de facto* standards like DWG. Some of the consequences were Product Models entailment to a high level of detail and communication of refined Manufacture specifications in a more complete and error prone way.

STEP became a successful standard and is making an immense contribution to the engineering activities. Aiming from the beginning to record both computational and human readable file, the purpose of human interpretation of data, after the physical file, is merely to enable slight adjustments or eventual typo mismatch corrections. Product geometry and part modeling features were not intended to engage standard parameters or variables with semantic interpretation properties. These are helpful to share additional information like design intent, the rationale behind features choice or catalogue details.

Thus, even after a physical file exchange, Design and Manufacture teams still need additional communication sessions to establish several unclear relationships of Product Model geometry, manufacture specifications or required adjustments. As an example, Figure 2 displays a portion of the 3D virtual model of the same product depicted in Figure 1, its technical drawing. The picture of Figure 2 was an excerpt from a Design - Manufacture teams communication, were the modification needed in the component was emphasized with simple messages annotated in a flat print screen of the component, guaranteeing adequate and effective human interpretation.

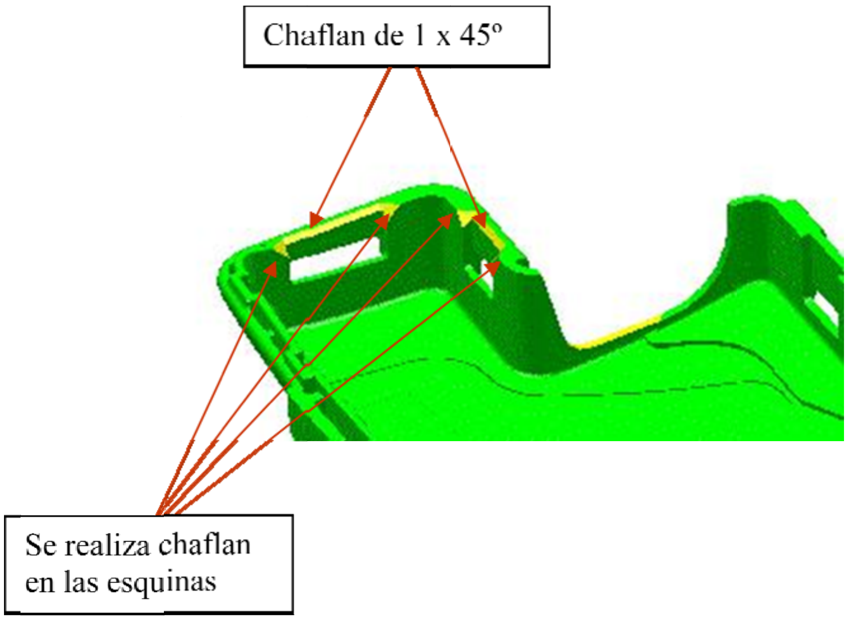


Figure 2 – Communication example between international Design and Manufacture teams – signaled required modification in a component (extracted from an industrial PowerPoint presentation – Spanish not translated on purpose).

Contrasting to the high precision of the geometric entities modeled in Product Models, other manufacture specifications were relegated to merely text based notes. The approaches used to develop computational Product models, like STEP standard product data models, privileged aspects of consistent geometry or well-structured features of Product data, rather than semantic configuration like standard variables required for knowledge capture and share.

For example, to easily report that a chamfer is required in a certain feature or that a hole diameter must be modified, an engineer has no possible association with the virtual

Product model entity. Further, a direct and simple evaluation of several nominal dimensions, based on the product model is still difficult or requires expertise, despite today computational ability. For instance, manufacturability validation still must be done in a human resources basis, connecting by hand computational systems that assist both design and manufacture.

In the reverse, disciplines like Design for Manufacturing, if granted with explicit and declared manufacture specifications, could perform automatic validation procedures during computational product modeling. To adequately establish inexpensive product specifications much depends on inherent Manufacture costs. A significant help would be brought to Design teams if subsequent activities could gave some feedback with valid and explicit information set, like manufacture specifications.

In a complementary perspective, within nowadays global market, Enterprises are becoming virtual organizations running almost fully computationally. Today, enterprises have information technology that could fulfill their requirements in each operational phase and with external partners, e.g., suppliers. For instance, in industrial environment, many applications are available to support operating Product Life Cycle stages.

An important phase of Product Life Cycle, in business perspective, is the Procurement activity which in the advent of internet era is having a strong improvement. Nevertheless, the available products information set shared in catalogues, like drawings and specifications, still present a broad heterogeneity at all levels, hence a lack of interoperability. For example, popular components used as fasteners, like screw and nuts, for long time are standardized. Depending of the involved partner (manufacturer, retailer, end user) is still possible to found in catalogues or technical reports variables nomenclature and features designation of a utmost heterogeneity.

Thus, it still is a challenging human task to a Design team to check if the component found corresponds to the specifications they need, or to a Manufacturing team re-use with liability stated specifications during process planning activities (e.g. at procurement or job shop activities).

The thesis will address these issues considering a point of view of nowadays product's lifecycle, and in the following section are explained the main line forces of the work.

I.2. Research question and directions

Research question

Can the harmonization of heterogeneous ontologies contribute to enrich integrated computational product models, making more complete the manufacture specifications in the support of a knowledge based Product Life Cycle?

From this question, the following research directions were defined:

Direction 1 – to analyze Product Model standards and its support of Product Life Cycle activities, identifying the reasons in the origin of flawed integration of manufacture specifications, specially in the lack of knowledge integration during the PLC;

Direction 2 – to identify nowadays Enterprise Modeling practices and barriers for a global integrated knowledge based PLC

Direction 3 – to develop a methodology to integrate knowledge through the harmonization of heterogeneous manufacture specifications in Product Models, contributing to reduce identified product model related interoperability problems.

Objective

And the objective of the Thesis was,

> Development of a novel methodology that facilitates the process of manufacture specifications integration in the underlying framework of product information - usually a computational product model - serving an integrated knowledge enriched Product Life Cycle.

I.3. Thesis Organization

This thesis is organized in five chapters, starting to describe the context and state-of-the-art, the challenges and the expected contribution, then the approach considered in the work is explained and finally details are given of the proposed methodology and its contribution.

Chapter I introduces the motivation of the work and focus the reader in the research directions in that the work was developed.

In **Chapter II**, entitled *Product Specifications in Product Life Cycle,* the state-of-the-art of Product Life Cycle, product modeling and product specifications definition contextualizes the reader with the theme and the identified issues.

The concept of product has been in evolution, and nowadays every simple artifact is a result of a complex chain with many intervenient sometimes spanned worldwide. Lifecycle of product has widened both in stages and detail. Industry needed to get adapted and prepared to such evolution, having computational resources an important responsibility in that objective. The standard product data models are described and several trends on product modeling extension, like parameterization capabilities, are presented. The chapter reflects a paper published "*Standard parametric product data representation: What's the STEP ahead?*"[2].

The **Chapter III** *Trends in Manufacturing Enterprises modeling* describes necessary background knowledge of Enterprise modeling, and exploits the concept of agents and multi – agents. Today's success of enterprises is based in communication tools inexistent a couple of years ago, and an important role has been played by computational agents. Guaranteeing the successful communication between business players, thus increasing the agility and responsiveness of the enterprises working in emerging virtual environments, is a key issue to keep manufacture competitiveness. It resulted from this research the publication "*Make the most of interoperability along PLC stages: A framework based on multilevel integration*"[3].

Chapter IV, *Semantic support to an integrated PLC* explains the usefulness of ontologies in enhancing the use and exchange of data, information, and knowledge among people and organizations, and in particular the semantic processing of manufacture specifications within distributed and heterogeneous systems and product models.

[2] in Proceedings book "Advanced design, management and production systems", ISBN 90 5809 524 6, CE2003, Balkema, July 2003.

[3] in ASME 2004 DETC/CIE proceedings, Salt Lake City , EUA September , 2004

Moreover, the distributed nature of ontology development has led to dissimilar ontologies for the same or overlapping domains, thus harmonization is required to keep systems interoperable. This contribution was fully published as "*Framework for Enhanced interoperability through Ontological Harmonization of Enterprise Product Models*" book chapter[4]. Two other contemporaneous papers detailed the proposal ("*Product Lifecycle Management Enhancement with an Ontological Approach*"[5] and "*Ontological harmonization of enterprise product models: an experimented scenario*"[6]).

In **Chapter V**, entitled *Framework for Ontology harmonization*, are discussed the pragmatic aspects of the research work contribution.

Even when enterprise models are interoperable, very often difficulties arise with respect to data semantics when information has to be exchanged, though common semantic models are not in place. When a standard mechanical component like a bolt is required to be selected by procurement teams or its manufacturability evaluated by Process Planning engineers, several issues arise lacking interoperability at different levels. The use of MENTOR as a methodology to develop Enterprise ontologies and respective harmonization pretends to contribute to assist the integration of applications and data. Two papers detailed thoroughly the proposal: "*MENTOR – A Methodology for Enterprise Reference Ontology Development*"[7] and "*Towards Ontology harmonization of Mechanical Manufacture Constraints through PLC*"[8].

This work is concluded in **Chapter VI**, *Conclusions and Future Research Trends* presenting a summary of work achievements and suggestions for future research related to this work. Main future research proposals consider the contribution of ongoing work in semantic STEP field and the extension of the specifications integration capability to Inspection phase.

---

[4] In Book chapter of "Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems, Sharman, R, et al, Association for Information Systems, Springer, 2007, ISBN 0-387-37019-6

[5] ASME 2005 DETC/CIE proceedings, Long Beach, California, EUA September , 2005

[6] IEEE 3rd International Conference on "Intelligent Systems" Westminster Sept, 4-6 2006

[7] IEEE IS'08 2008 Fourth International IEEE Conference on Intelligent Systems, Golden Sands resort, Varna, Bulgaria, September 6-8, 2008

[8] Proceedings of COBEM, 20th International Congress of Mechanical Engineering, November 15-20, 2009, Gramado, RS, Brazil

Chapter II

II. Product Specifications in Product Life Cycle

During the last years, industry has observed the evolution of the Product Life Cycle (PLC) concept. Each industry simultaneously belonged to a complex network and contributed to growing density of such network, as a consequence of highly structured interaction maintained with other partners. As a result, the core business specificity of each industrial partner contributed to the enlargement of the Product Life Cycle concept itself [1]. Conventional design and manufacture stages recently gave place to activities with narrowing scope in PLC (where early Virtual Product marketing or specific product's Maintenance or Disposal activities are examples).

By the other side, final consumers are requesting more features in traditional goods, and are looking for customized products which availability requires production scheduling reserved several years ago for large production series. Such evolution implies thousands of parts correctly ordered, manufactured and assembled, and implies the involvement of complementary activities and experts from a wide range of disciplines and expertise [1], quite often geographically distributed. Nowadays, requirements like error minimization, product cost reduction, delivery time shortening, and product quality improvement [2], [3] are exigencies almost mandatory in each context.

In a mix of need and consequence, the conventional way of developing a product, involving sequential stages from various specialized working groups become replaced by concurrent and collaborative engineering methods [3] [4] [5]. This innovative environment of collaborative tools revealed to be decisive to keep competitiveness of most industries, from the biggest like automotive and aeronautics to those strongly involving SMEs, like furniture or building and construction [2] [5] [6].

In recent times, the computational refinements achieved in product modeling and virtualization, numerical control both in manufacturing and inspection, ERP systems and others fields enabled the evolution of PLC into business orientation. *Product Lifecycle Management* concept emerged with the promise of integration of data, information and knowledge required to a successful product and production process and at the same time availability (in time and format) to all involved partners (both from supplier and client sides) [7].

The advent of Internet enabled the e-commerce as a powerful economic tool to modern PLC and e-Procurement has become an important promise of standard and tailored products almost accessible of the shelf [8] [9].

An analysis through the above contributes will focus the thesis theme and perspective the scenario in which some contribution is made with the present work.

II.1. Evolution of Product Model as an Information Vessel

The evolution of concepts like the Product Life Cycle as well as the Product Lifecycle Management will give the reader a better insight of the mismatches and bottlenecks still existent in the computational integration of manufacture specifications. In the end of the Chapter are explained the standards available in manufacture field to represent product information and the trends towards manufacture knowledge capture.

II.1.1. Evolution of Product Life Cycle concept

The Product Life Cycle concept and definition may vary according different scientific disciplines and has been under attention of academic and industrial community. In a sentence, Product Life Cycle comprehends the stages through each artifact evolves over time from first thoughts and ideas (thus considering the very beginning of the product) until the last function or utility that a product may have (even beyond its physical existence as a whole, which extends the concept to ultimate stages of components' disposal after disassembly). Stages considered in such cycle may vary, since Financial or Economical point of views are less ambitious than Engineering, both in number or scope. Moreover, although a macro perspective of its major phases was sufficient several years ago to give an overall picture of product's lifecycle, it needed to be re-configured because the PLC concept broadened and refinements needed to be introduced due to new fangled activities (from early conceptual design like product market surveillance to final disposal activities like recycling analysis).

During the last decade, many proposals were developed to model an overall schema of *product life cycle* activities. The work of Hollins & Pugh [10], as an initial reference work, exposes a clear identification and with sufficient detail, from the schematic point of view, of each phase, stage and function, leading to a successful final product.

Typically, the complete product development and deployment is divided into major phases [2] [3] [11], and within each phase specific activities classify the evolution of the product in several stages. This division has not been always consensual since major areas like design and manufacturing are in constant evolution, embracing different knowledge background and domain expertise [4] [12] [13].

Moreover, the introduction of new methodologies for production efficiency increase, particularly in the case of concurrent and collaborative techniques, enabled the clear division in a significant number of identifiable Product Life Cycle (PLC) stages and activities.

Companies witnessed the traditional path of production stages giving place to the new organization of simultaneous stages of design and manufacture, and introduction of *Design for X* disciplines, like Maintenance or Recycling. The work of Chris Jones lists dozens of design methods [14]. In parallel, this collaborative environment became powered due to computational skills that the PLC processes meanwhile acquired. The advent of the networked *computational era* brought appropriate software tools, suitable to assist the execution and interconnection of the PLC production processes. This migration process from traditional to computational tools was inevitable and currently is still ongoing at companies and production chains.

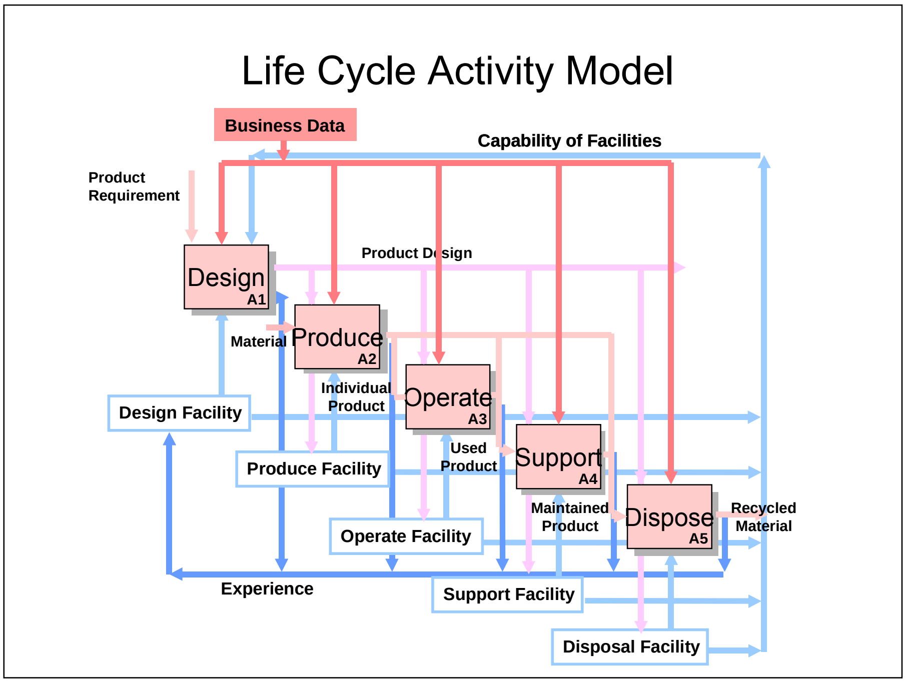


Figure 3 – Main division of PLC activities proposed by ISO TC 184/SC 4 (from [15]).

Thus, PLC activities are having growing attention and its modeling and study is ongoing by several communities [16] [17]. For instance, supervision on information flow and in interoperability of computational data lead ISO TC 184/SC 4 to thoroughly identify major phases and activities related to product lifecycle[9] (see Figure 3) [15]. With similar purpose, the Society of Manufacturing Engineers made his contribution, organizing the configuration of industrial activities and made a compilation in a form of matrix of PLC phases.

To assist today's production chains a new set of computational resources is available, and its identification re-designs the PLC schema. As a consequence, the Pugh schematic level does not detail these all specific activities computationally based in the PLC schema. Several recent contributions from standardization bodies like some *ISO Working Group 10* or *Society for Manufacturing Engineers* were resumed at Silva's work [18]. A new scenario has become available that deals with fully computational product information and data along PLC, with tools and applications addressing purely computational product data (see right side boxes of Figure 4).

[9] Despite a glossary has been published in an ISO TC184/SC 4 document (N326) no hierarchical distinction is made in terms of **phase**, **stage** or **activity**. According to (N326), Product Lifecycle activities are divided in 8 big categories, named as "Phases", and several activities occur inside these major divisions. Although the document has no purpose on the term's definition itself, a glossary is included in a separate chapter, which confirms that terms sometimes become mistaken, misused or with unclear semantic definition.

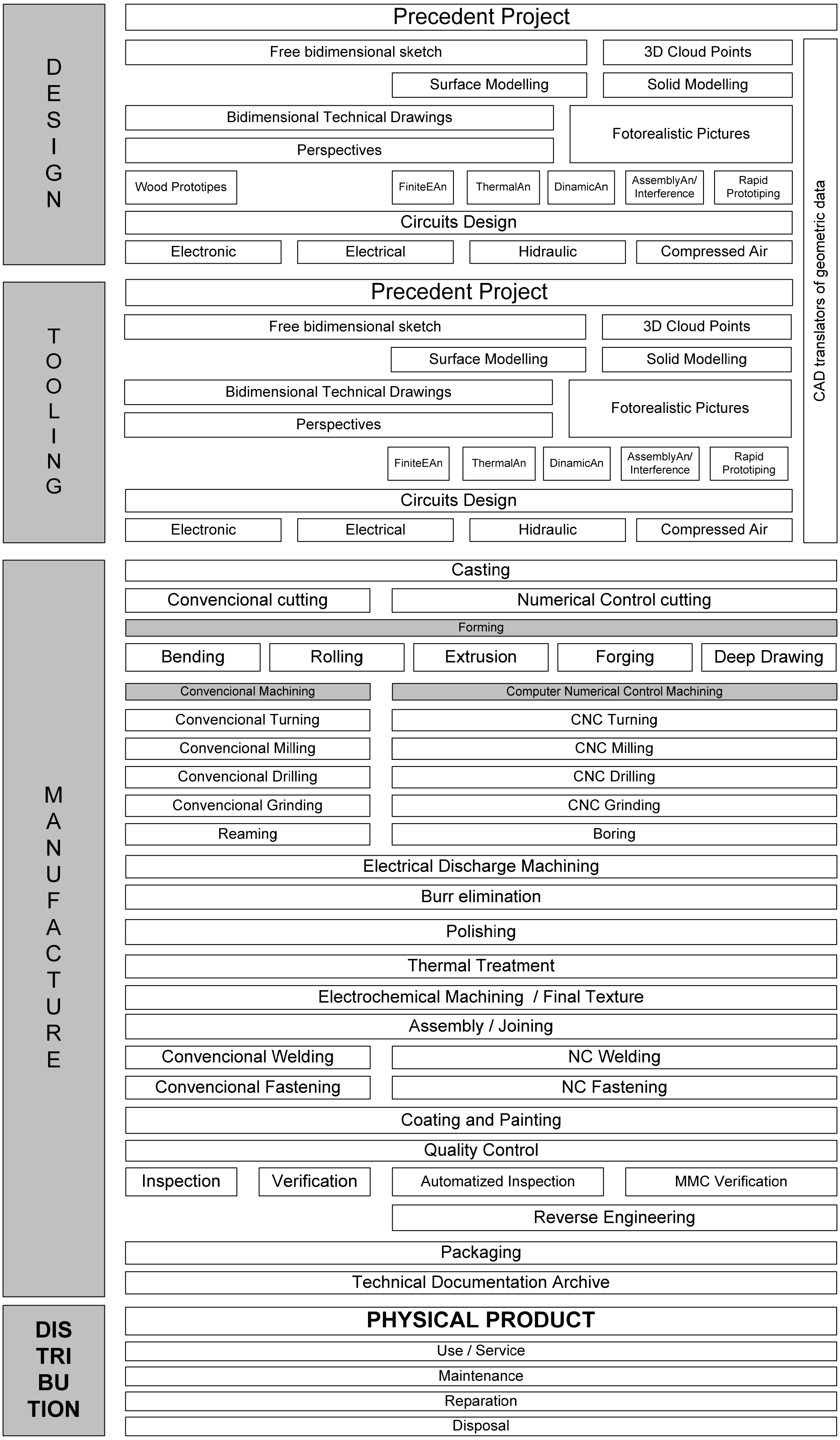


Figure 4 – Traditional and computational Product Life Cycles phases and stages identification [19].

The activities involved in PLC are vast enough both in quantity and variety. Formally, **lifecycle activities** are defined as *the set of tasks executed during the life of a product (the design, production, operation, support and disposal of a product)* [15]. These activities, that make up the lifecycle, vary from one industry to another, and their relative importance too. Whatever the industry, alone or in association with others, each activity fits in the above or a similar schema. On the other side, the product itself, embodied towards his final shape along these activities, may be considered the physical stakeholder connecting all the stages –the material flow. Amid this flow, it is carried out the information or data stream, feeding each stage with necessary product and process data. With prosperity, dissemination and helpful skills of computational data support, it was unavoidable it's introduction in production chains assisting those tasks.

Depending on the extents that computational level is implemented in each company, the available computational tools that assist nowadays industrial workforce are, in part, responsible for the fully migration from traditional to computational product information and data support, seeking to digitally connect software tools of early design stages with the final production chain activities, like manufacturing inspection devices.

II.1.2. The Product Lifecycle Management concept

The Product Life Cycle concept extension beyond the main phases of design and manufacture lead to a wide-ranging point of view of the product's lifecycle. Appearing as a new paradigm for manufacturing companies, at first, rapidly the *Product Lifecycle Management* has generalized the concept of managing a company's product all the way across its lifetime in the most effective way and to every kind of industrial sectors.

For various reasons, the concept only emerged after nearly two decades of market and technological evolution, in the middle of the 1990's. The cadence of new technology's introduction in the 1980's has not contributed to the concept consolidation, and the blossoming of a large set of disciplines (and acronyms) neither. Frequently, popular issues like what product and process information should be under PLM control or which disciplines to involve in the PLM (engineering, business or many others) where also an obstacle in a certain way to the consolidation of the concept [20].

Only over the time, more and more successful and proving examples enabled the clear wide-ranging scope of PLM solutions. Figure 5 illustrates the perception of successful cases in the Business staff over recent years. If the first implementations were concerned on engineering drawings or early design data, the latest PLM solutions provide best practices of business processes, with focus on complete business solutions that address top and bottom line issues [7]. Basic data from engineering area turn more and more connected and related with business processes.

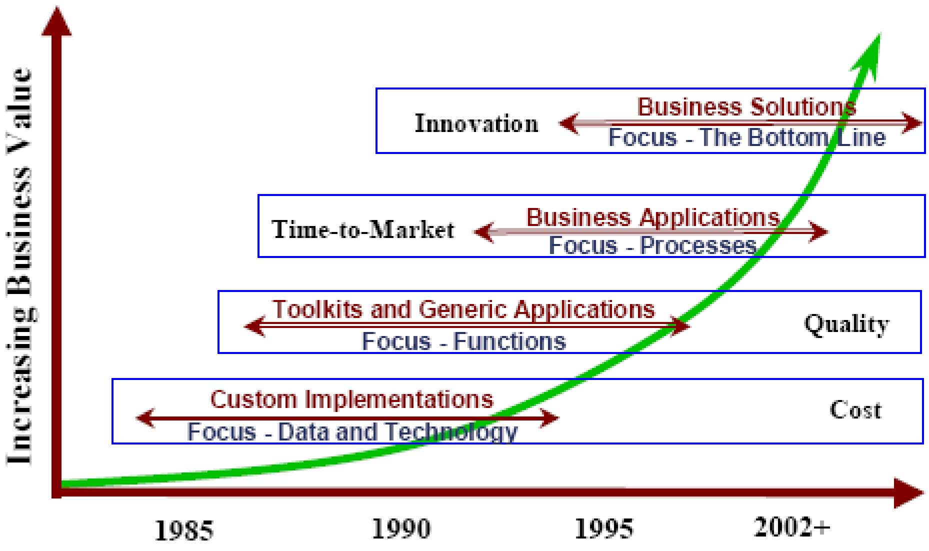


Figure 5 – Evolution of Product Lifecycle Management scope, according to CIMdata [7].

The introduction of *Product Lifecycle Management (PLM) concept* refreshed the promise of seamlessly integrating all the information produced throughout all phases of a product's life cycle to everyone in an organization at every managerial and technical level, along with key suppliers and customers [21]. In parallel, *PLM systems* stand for the [computational] tools constellation that implements the PLM concept. As a consequence, the capability to serve up required information to PLC actors, the cohesion and traceability of product data, among others, become properties with capital importance in PLM systems.

Today's retrospective of early CAD/CAM systems or rudimentary ERP applications reveals that computer support to Product development was focused in a segmented part of a product's lifecycle, typically the product's engineering specification, its physical embodiment or the basic business data management. The natural evolution was to gather all information and data related to the product in such a system that every phase might be supplied with the right piece in the right time. The CAD geometric data addressed by neutral translators in early design stages (see Figure 4) needed then to be extended in contents and to the subsequent stages of PLC.

In terms of software support, several functions were explicitly identified as top priority on the requirements of early PLM systems support. Among others, the following should be considered:

- Translation of design data to manufacturing;
- Production operations planning and machining process planning;
- Assembly definition and sequencing;
- Aid on process planning;
- Detailed line, cell, station, and task design;
- Quality measurement and reporting;
- Manufacturing documentation, shop floor instruction, and collaboration schemas;
- Standardization of information and dataflow.

Product Data Management systems, the evolution of advanced CAD systems, were the subsequent software tool for collecting, storing, organizing, managing and making accessible product and process information. It is proposed as a set of software tools designed to control and electronically simulate a product throughout its life-cycle.

PDM systems are considered among the most important components of a PLM solution [20]-pp.233]. As a primary system and a visible component of a PLM solution, a *PDM system* have the main responsibility to manage product data and product workflow.

Today, the aspects of a PDM system that reveal to influence its performance in a company are those addressing information and data issues. Typically, the most relevant components which must be carefully reviewed before acquisition of a PDM system are:

- the information warehouse (where product data is stored),
- the information management module (manages the information vault, both its content and context),
- the user interface, interfaces with other systems (interoperability),
- the information and workflow structure definition functions (to define the structure of the data, content of documents and technical publications, to manage product portfolios, engineering changes and component supplier management); and
- the system administration (to assign access rights and information authoring) [7] [11] [20].

Therefore, under PLM paradigm, a proportional increase of the volume, diversity, and complexity of information was expected, describing the product and the process, which systems must address [21]. Silva identified several pieces of information of a standard component, feeding different data to some PLC stages (schematic shown in Figure 6) [18].

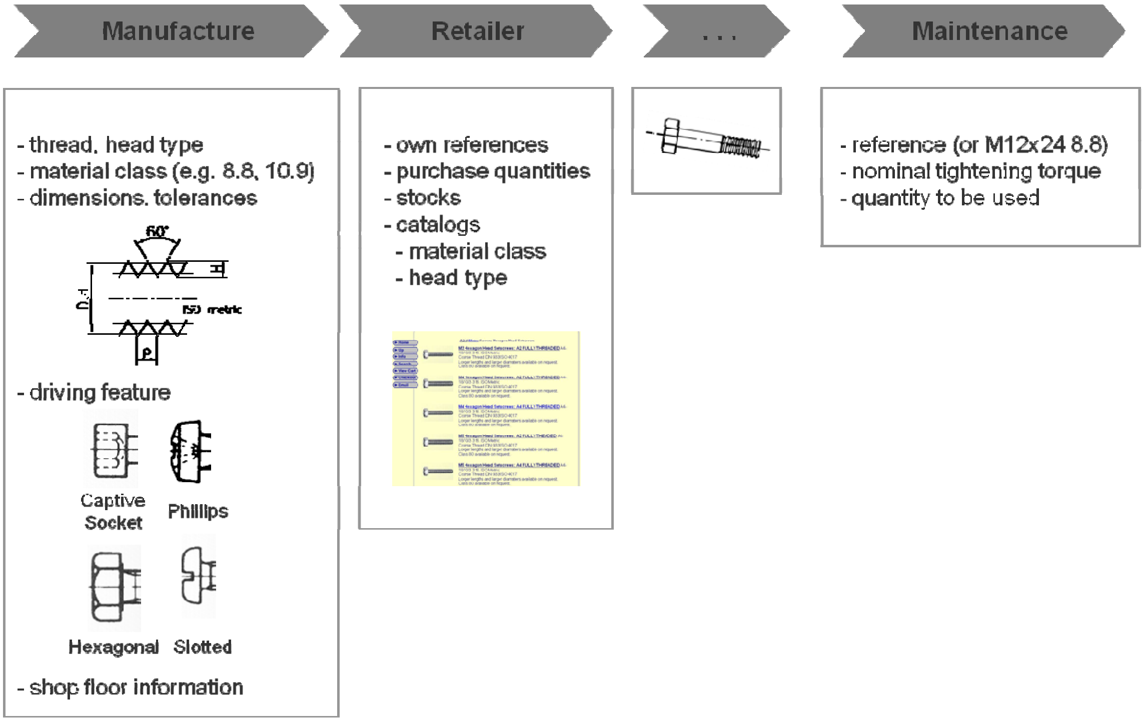


Figure 6 – Product's data items required in different product lifecycle phases.

The attention and concern on information and data quickly came under focus of major software vendors, namely since many issues were unveiled when systems needed to communicate between each other. A robust framework was then necessary to provide information support for the product's entire lifecycle, from the first conceptualization to the disposal of its last instance. Design, Manufacture and Maintenance all work with the bill of materials (BOM), the list of the parts and subassemblies that make a product. This common place and data base was an initial draft of what was necessary to gather, to share and to turn integrated [20]. *The model of the information carrier was by that time emerging as a real issue to be addressed as quickly as possible.* The information and data modeling become a requirement needed to be suitably addressed in order to keep systems working properly and efficiently. In fact, even with non guaranteed return-of-investment, companies invested in PLM systems after the belief that PLM systems would bring them a satisfactory information management and consequent competitive key to face the market dynamism [22].

Manufacturability

The particular phase of Manufacture begun to claim support to process planning, factory modeling, visualization and collaboration, simulation of operations, ergonomic and human factor analyses, and other tools necessary to achieve integration and optimization of the manufacturing process.

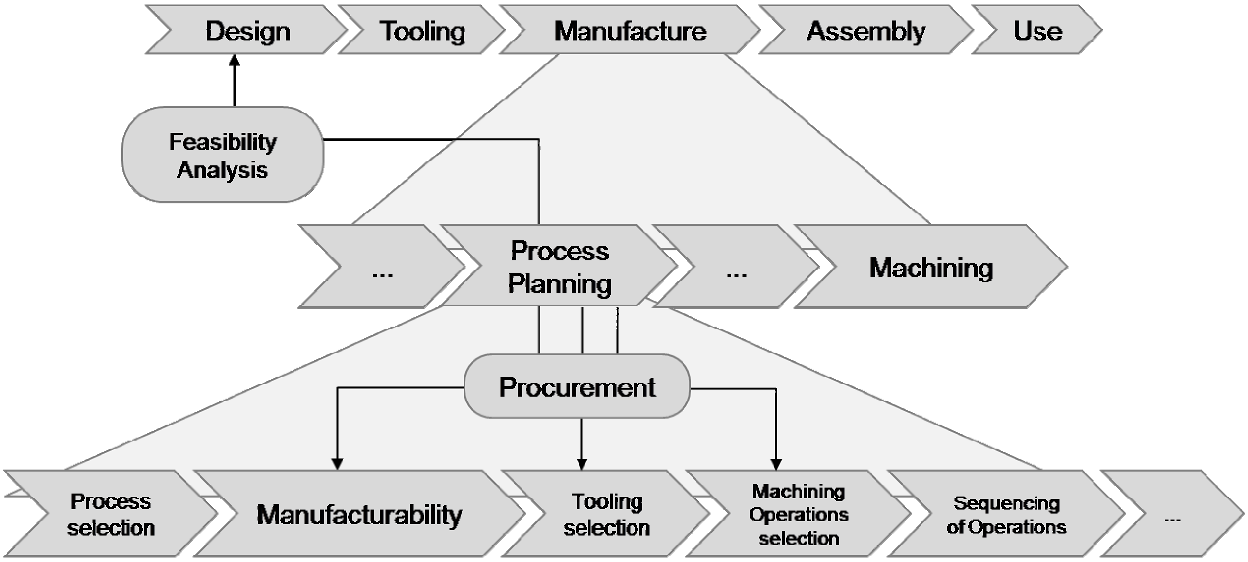


Figure 7 – The need of manufacturability information in the product development process.

An example will describe the scenario under which current work was developed. Whatever approach each industry uses in its product's design, important information and knowledge is required during these early stages of product's lifecycle. As seen in Figure 4, multiple processes are involved in every product's manufacture, and contribution of productive process specifications, inherent to each specific stage, which results are condensed in the final product properties. Process Planning is the PLC activity responsible for such process definition and occurs traditionally after an initial iteration of product's design. As seen in Figure 7, the manufacturability activity will check the feasibility of the component under the selected process. In general terms, several approaches were proposed to ensure that every life cycle activity is provided with formal manufacturability information in time: Concurrent Engineering, full range of Design for X principles, PLM itself among others. The benefits of each one were extensively described in literature [17], [21] [22]. Nevertheless, despite the computational support enabling the easy share of such information, design stages still lack of adequate supply of formal manufacturability information [16].

In parallel with the materialization of the product itself, the Manufacture stage produces a complete set of product information, nowadays fully computational in certain cases. Ironically, such powerful resource scarcely is re-used in a convenient and profitable way. Only after several iterations, an engineer in the field becomes accustomed with the manufacture process and then masters the required manufacturability information to be used in new product's design.

The problem of informal manufacturability information used in the early phases of the product development process is that untrustworthy feasibility studies or unnecessary design iterations may occur, with consequent loss of productivity and market positioning. Although several approaches were proposed, it was identified that availability of up-to-date, high quality information is often unmet and decision makers still have to rely on incomplete, low quality, or inconsistent information [23], or on personal experience. The answer to a complex problem, such is the information feeding of design process, is still open. This information re-use issue will be addressed later in this work.

e-Procurement

Similar to wasted or unused Manufacture phase's information set (shared back to precedent stages), the same can be verified in subsequent product's lifecycle phases. One important activity that assists Process Planning (see Figure 7) is the Procurement stage, both for materials, parts or tools. Procurement, in a PLC perspective, may be seen as the stage where information of the manufacture good is re-used. The properties of the desired component are verified, matching the desired shapes, functions and other specifications that will be required. In today's scenario, the component under selection may still physically be non-existent at that stage. Under PLM also this concept has suffered a great evolution, with products being virtually displayed without physical existence and its manufacture being a last minute order.

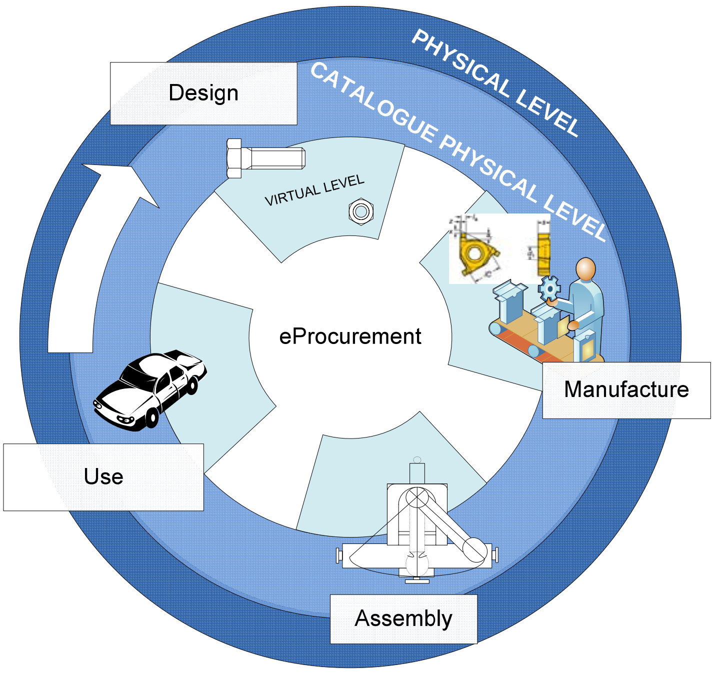


Figure 8 – Level's evolution of catalogue product data available to e.g. nowadays eProcurement activity.

Figure 8 graphically resumes the role of Procurement in PLC, showing the evolution from the physical level as it was used to happen several years ago to a complete virtual environment in a near future. The white sliced inner part of the circle represents the role that Procurement activity has between each main PLC phase displayed. Each part may be seen as a final stage of different productive lines, which will be incorporated in the bigger system that is ahead in relation of its lifecycle.

A well-known tool traditionally aids Procurement activity: the catalogues.

Catalogues are very widely used throughout the industry for procurement support to Design, Tooling, Service and Maintenance activities across the whole product lifecycle. The catalogue of a product may be seen as a small-scale repository of product's information: characteristics, functions and performances of product may be evaluated after catalogue's data.

Among a myriad of other possibilities, part of the information set typically comprehends: the summary of technical data; setup and service information; design, construction, material design data; details of components; product picture; sectional drawing and nomenclature; graph showing product performance limits; a series of product's characteristic curves (graphs), product dimension sheet (drawings), and drawings showing interconnecting components, possible arrangements and product options.

If a decade ago paper based catalogues were the main support to this activity, recent computational tools enabled the electronic publication[10].

Ideally, every equipment supplier should maintain and distribute its equipment data in an electronic form that is acceptable and usable to all the companies that are its customers. More recently, PLIB has assisting the modeling of the information of product libraries and catalogue data, which has been of precious help in this field [24].

Driving PLM evolution is the increasing complexity of products and the growing number of requirements of the clients. Several parts need to be checked more than once during design phases, or later in the product lifecycle like maintenance, and electronic catalogues are the most effective way (e.g. with aid of product virtual models, like CAD models). Many companies realized that the maintenance of up-to-date and consistent data catalogues, despite its difficulty and cost, was an important process to the commercial

[10] Nevertheless, there were times where few catalogues were exchanged in electronic form, and then were nearly always in proprietary formats (CIMdata).

success of their products. In the other side, it could be unwise to publicize out-of-date catalogues, since they will be the origin of many issues, namely those related to design purposes due to implied and inducing errors as noted before.

Finally, PLM systems emerged induced by the many changes in the environment for manufactured products. One of these was the rapidly increasing number and complexity of extended enterprises made up of a manufacturer and a chain of suppliers. Automotive industry was an example in the area, but the contamination of the concept to other industries, particularly in SME sector, was predictable. The advent of Internet enabled the extended enterprises to use e-commerce as a powerful economic tool in PLC context and e-Procurement has become an important promise to computationally assist every PLC phase. Tailored products almost accessible of the shelf, powerful product visualizations even before its materialization, knowledge capabilities integration are some of expected shifts that e-Procurement is currently introducing in the electronic market [8] [25].

Therefore, the share of catalogue information with every PLC phase in an adequate manner is of great importance to enterprises and to setup PLM concepts in advent of the new computational era.

In resume, tackling properly the information about the *product* has become essential to avoid several integration and optimization obstacle issues that still occur in several products' lifecycle stages.

II.1.3. The Product Model as Enterprise Information stakeholder

Information exchange between internal and external agents, both referent to the product and process always has been the core engine of successful Product and efficient Production units [26]. Moreover, to fulfill concurrent design principles, integration capabilities of a suitable information carrier became more and more requested in the early stages of the product design, and at several levels (technical, organizational, among others). Under the new PLC architecture and PLM concepts, with determinant time-consuming tasks in a successful product in the market, the management of such information and knowledge inside corporations assume a crucial relevance.

In a company, this information and expertise may be structured in three major interacting divisions: Product definition, Production definition and Operational support. The first two are within the scope of present work, with focus on product and process model of the information and data flow in the PLC; the later refers to business and market aspects that are not formally addressed in present work.

Product Model

Due to rapid development of information technology, product models have been proposed to integrate the various information aspects about products [27] [28] [29]. Product modeling has been recognized by academic and industrial communities over the past several years, together with an advanced information system, the best way to integrate and co-ordinate the various lifecycle considerations during product realization. Virtual product models like Computer-Aided Design, for generating a product's geometry, and Computer-Aided Engineering, for analyses of its behavior, are quite in common use in industrial environments. Typically, a product's behavior needs to be analyzed in several functional domains (e.g. structural, thermal, kinetics, economical) before its physical existence. The behavior, improvement and optimization of the design changes are more and more relying on product model analysis.

The effective support to Design stages – conceptual, detailed, etc - , requires that a product model should have at least the following characteristics [28]:

- ability to capture information from conceptual phases along with the information from the detailed design stages;
- the information should be organized in a way that is easy for the users to access, which means that the information structure should bear close resemblance to the user's thought patterns;
- an unbiased generation of possible solution alternatives: given a set of functional requirements, the system should not only be able to offer the user as many design alternatives as possible, but also provide enough detailed information of each alternative for objective evaluation.
- the information should be organized hierarchically so that the designers can search design alternatives at different levels of abstraction;
- the information should accommodate a structure that can give effective support to different scientific disciplines to be explored in a model analysis: mathematical, physical, numerical, geometrical are just some examples.

In the course of product development, a considerable amount of information and data is produced and processed at various places in a company. Quotidian activities inside Design and Manufacture phases exchange large volumes of product information, and a product model needs to support such 'horizontal' information exchanges as suitably as the 'vertical' exchanges among other components of PLM systems and intermediary systems, such as CAD, Product Data Management and Enterprise Resource Planning (ERP) systems [21].

For the PLM concept to be successful, issues such as management of large-scale data (expected from the exhaustive description and representation of the Product) needed to be addressed so that information can be provided to individuals in a useful way. In addition, companies needed settlement of data standards and design of corporation-wide integration architectures in order to keep data and information under control [7] [30]. To make the most of PLM systems, people at different levels inside companies should be able to make key decisions – like to prevent unfeasible components' features with available machines or to determine shop floor tolerances range – when they are most cost-effective, rather than midstream in the manufacturing or even during finishing stage.

Several proposals for a PLM system support framework for product model that can access, store, serve, and reuse all the product information throughout the entire lifecycle were developed [21]. *Also,* the IMS (Integrated Manufacturing System) organization proposed the use of a system that recognizes and feeds computational information independently to each phase of the manufacturing cycle while at the same time maintaining a database that serves as a single source of data for all company activities and applications. Basic data was maintained in current and accurate condition so that information can be provided on demand.

If in traditional PLC principles, with mostly human based tasks, the Technical Drawing were the best appropriate tool found to carry required PLC information, a more suitable product model needed to be adopted.

For instance, it must be acknowledged that, as product design decisions are made, they should be immediately evaluated in terms of primary aspects like functionality or manufacturability cost. This requires sharing of a unified product representation as a reference model to integrate and coordinate the life cycle tasks along production chain activities.

Additionally, depending on the PLC stage, different modular views of the product model are addressed. The instantiation of an initial product model will be sufficient if containing knowledge and information regarding to the early design and tooling phases.

However, this model needs to be updated along the whole PLC, for example with additional information for manufacturing, inspection, use or maintenance stages.

To exemplify, the selection of a sequence for production of a desired product is based on the knowledge of the capabilities of the various manufacturing processes along PLC stages, including product material requirements, shape complexity, surface finishing and tolerance exigencies. Thus, a computational support to exchange data and carry information along PLC stages, bridging the intervenient stages with knowledge integration ability, thoroughly enhance the performance of the product model data.

Computer aided design (CAD) tools were increasingly adopted by modern companies during last decade, enhancing high quality product model representation and definition. CAD technology currently plays a key role in today's advanced manufacturing environment [31], in particular due to the widespread product representation they use, superseding traditional technical drawings and paper support. CAD has becoming a vital product model platform in industry, and design teams increasingly depend on various CAD applications, such as CATIA, Pro/E, AutoCAD or UGII.

Nevertheless, CAD systems do not include appropriate data structures to accommodate complete PLC information. Beyond an accurate geometric model of the product, information regarding tolerances, surface finishing, design intent or technical constraints are typically not included in current geometric CAD models, being simply represented as text labels on the drawing. The text format is also adopted in technical notes and some product specifications.

Traditional product modeling techniques and computer-aided design systems are mainly focused on physical product modeling and geometric representation, which makes them insufficient to help in some PLC stages, like conceptual design, and to properly deal with product requirements and product specifications.

Therefore, when a product model is transferred to downstream users such as the process planner or the inspection planner, users repeatedly need to regenerate from the text label the necessary manufacturing information through human intervention. To avoid this inefficiency, an integrated product model should be considered, in which all sort of information such as geometrical data and manufacturing specifications could be stored together and shared. The STandard for the Exchange of Product (STEP) data model, which is defined as the international standard ISO 10303 [32], has been developed to enable not only geometric data sharing, but also technical and management information convey. What is more, the standard enables the adoption of partial reference modules, which has been proven as a significant contribution towards a solution for the product specifications share between PLC stages.

Other non-standard based contributions have been made to extend Product Model capabilities. For example, Tay & Gu presented a product model for developing conceptual design support systems [28]. The proposed model captures not only geometrical information, but also non geometrical information, such as function and constraints. The implemented prototype conceptual design support system organizes basic information items of a relational database and geometrical modeling capabilities after a conventional AutoCAD drawing.

However, designers should concentrate their attention on the product itself, rather than on the product model to be adopted. Regardless designer's know-how of the production processes to be adopted in manufacture, assembly or supply of a product, the main element determining the failure or success of the product is design knowledge, that is the base and core factor leading to competitive products. Without a robust support of design intents (and knowledge resources), any kind of product design method or developing mechanism will be unsustainable within nowadays marketplace [12].

Additionally, due to the increasing product complexity, it is unavoidable for a product designer to utilize in some extent various knowledge resources to develop his product. Therefore, it is not sufficient that the product model completely represents or depicts the product shape or properties. Knowledge incorporation capabilities should be included and made accessible along PLC stages.

The progress made through the adoption of standards, such as STEP [12], also reaches a compromise solution. Nevertheless, proper approach to knowledge incorporation in current STEP product data model is an important issue to further improve product model integration [33].

II.2. Standard Product model: requirements and manufacture specifications

In order to integrate a Product model with Product Life Cycle stages, developments achieved within STEP community have promising results in the point of view of support for the exchange of product data. ISO 10303 STEP is a product modeling approach that enables all the aspects of a product development project to be taken into account (and data), so contributing to product model integration.

II.2.1. STEP – a Product Model Exchange Standard

In 1980, The US National Bureau of Standards published IGES – Initial Graphics for the Exchange of Product data). The standard was developed on purpose focused on graphics, with a simple approach to define the data representation format. ISO 10303, under the general title of industrial automation systems and integration-product data representation and exchange, is an international standard for the computer-interpretable representation and exchange of product data (STEP). STEP was intended to be an improvement upon IGES in the following respects:

- to provide a data model; and
- to encompass a larger product structure than the shape.

The ISO 10303 STEP standard provides a mechanism that is capable of describing product data throughout the life cycle of a product, and is independent of any particular system [32]. STEP was formally accepted as an international standard in 1994. It was the first standard concerning complete product model exchange, and is also a platform and methodology for the implementation of object-oriented software development.

It enables consistent implementations across multiple applications and systems, and is suitable not only for neutral file exchange and application programming interfaces, but also as a basis for implementing and sharing product databases. The STEP standard covers a large scope and deals with highly abstract product modeling information. Therefore, in order to speed up standardization, STEP is organized as a series of parts, and each is extended and published separately. All of the parts belong to one of the following series:

1. Description methods
2. Integrated resources
3. Application protocols
4. Abstract test suites
5. Implementation methods
6. Conformance testing

STEP product modeling is based on integrated resources. Integrated resources, which consist of generic resources and application resources, define a generic information model for product data. As an element of STEP, an application protocol includes the definitions of scope, context, and information requirements of an application. It provides the capability of interpreting the integrated resources to meet the product information requirements of specific applications. The application activity model (AAM), the application reference model (ARM), and the application-interpreted model (AIM) are the three important resulting models documented in an informative annex to the application protocol. A mapping from the information requirements to the AIM is also provided within an application protocol. Examples of application protocol are the AP214 for the automotive mechanical design processes and AP203 for support of configuration control. In Annex I is listed an example of the STEP Part 21 file of a mechanical component, using the AP 214.

In order to support the development of integrated product models, STEP uses a formal information modeling language named EXPRESS, which is itself a part of STEP standard (Part 11).

The product data in integrated resources and application protocols are described by EXPRESS to ensure consistency and avoid ambiguity. The graphical representation of EXPRESS is EXPRESS-G, which is provided to aid in understanding the definitions and relationships established.

Compared with the previous data exchange standards such as IGES and SET, STEP provides a product modeling approach in which all the aspects concerning product development life cycle, including organizational data and geometry, are taken into account (see Annex I). At present, STEP is still a developing standard. Many companies around the world are supporting this industrial standard, with a wide range of commercial CAD/CAM and CAE applications STEP compliant.

Parametrics

More recently, parametric capabilities have been incorporated, since such features provide STEP product model with parameters representing other product information like material properties or technology and manufacturing properties. In a CAD system, after modeling the geometry of a product or component, much more information complementary to final geometry can be shared, like the history of features and their behavioral relationship.

In the information exchange, not only representation (ad thus data file) results simplified but also extended information is included. For instance, product constraints, which can be subdivided into *geometric constraints* (adding parameter to geometric coordinates between topological objects and respective constraints) and so called *engineering constraints*, relating geometric and other product properties. Examples of engineering constraints are parameters like dimensions correlation, material strength or machining variables such as cutting speed or federate. These parameters can be associated using arithmetical (functional) or logical expressions, thus adding engineering knowledge to the product description.

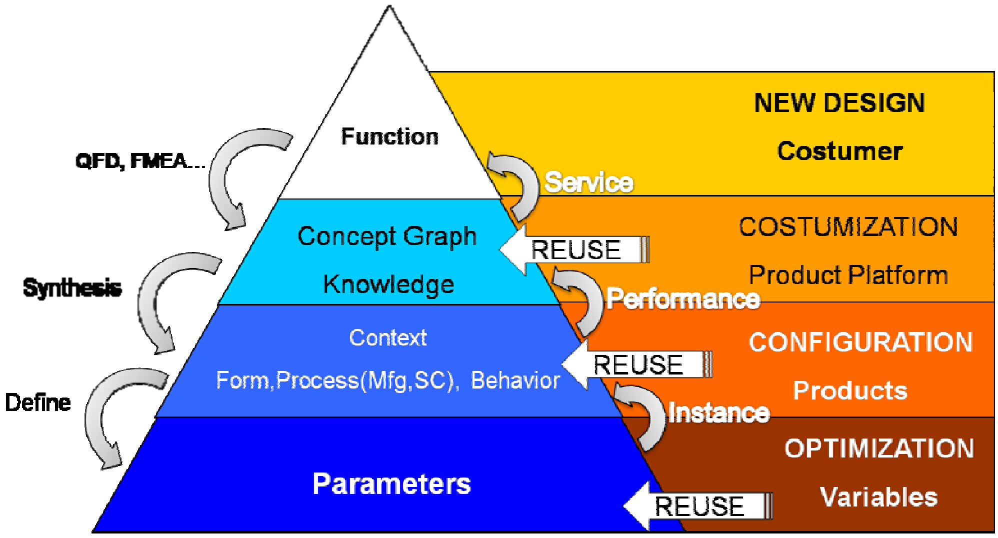


Figure 9 – Correlation between Function and Parameterization of a Product Model during Design Process.

One of the objectives of parameterization was to mirror the functional aspect (that components necessarily have) in STEP product data model (see Figure 9). The need for rapid adaptations to changing customer requirements, the requirements increase on product properties and the general pressure of costs recommends, as much as possible, the re-use of data and information of precedent phases and iterations. In fact, conceptual design proposes an initial iteration that subsequent PLC phases will update with slight changes, and thus parameterization enables instantaneous and automatic updates. Error-prone and repetitive tasks are this way prevented to Design teams, freeing time to more important tasks.

Although great advantages may result of such enhancement, and product's knowledge could be captured to some extent, parameters have some limitations as product specifications carrier.

In the particular example of Figure 10, the two cylindrical surfaces (i.e., preceding thread surface) may be produced accordingly the way they were designed. One hole will be resultant of a milling operation which generates the planar surface of the bottom, and the other enables a drilling operation, giving a conical surface to the bottom.

Parameterization of both features may enable properties to be easily and quickly changed, however, feature / geometry doesn't change in terms of manufacturability impact, but in terms of product specification. By other words, those two holes as pictured have the geometrical specification restricting the manufacture process, and parameterization will not carry the necessary manufacturability information about possibility of feature alternative.

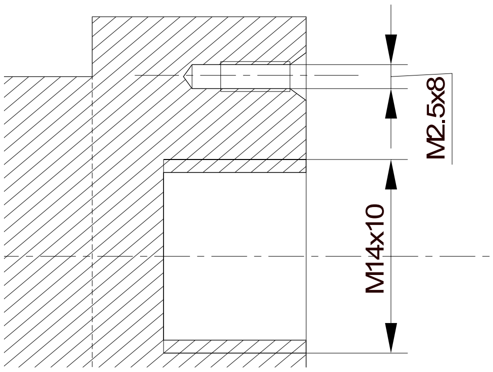


Figure 10 – Different hole features specified by Design constraining the workpiece Manufacture.

Therefore, the manufacture team may be prevented to seek cost reduction keeping component functionality unchanged, e.g. drilling the larger hole, an operation with lower process cost than milling operation.

II.2.2. Manufacture specifications: human and computational proficiency

Product specifications

A Product specification means the precise description of what the product has to do and accomplish [4]. Depending on the type of Production Company, product specifications may be under the responsibility of the client, of the supplier or shared by both. By the other hand, small series companies have a Design Department more committed to discuss dynamic product specifications with their clients, while Process industries usually have invariable and monotone product specifications.

Manufacture specifications is the subset of the product specifications carrying information needed to a successful product's manufacture. Often they are neither distinguishable among others nor classified as such but, surely, they play a decisive role during Manufacture PLC phase (and the proof happens when they are missing).

A general development trend observed in modern companies is a desire to move towards mass customization: customized products offered at low prices (further details in following Chapter[11]). Although the first goals to be met were reducing costs and delivery times, currently industry and research community struggle for the improvement of specification processes. Moreover, costs of manufacturing process are intrinsically correlated with the specifications made during quoting process. A typical problem in relation to this is that salesmen often promise special solutions which later prove to be very costly, just to generate revenue. A better insight into these consequences makes it possible to make more optimized decisions early in the process.

Figure 11 sketches a simplified overview of Product Life Cycle information flow in what product specifications respects. The entry point may be considered the Product Idea (pushed by marketing or any other client initiative). Countless documents are involved in the product's lifecycle (details further in the continuation) but they are not fully represented on purpose. The window in the middle puts in evidence the documents (whatever they are in physical or electronic format) that are involved in client and supplier communication process when a New Product design initiates. From left to right, it can be seen that several entities are combined to refine the properties that the product should have (or may have, due to available manufacture capability). The initial specifications set commonly agreed between client and supplier are then manufactured and service and catalogue data becomes available – that corresponds of data included in service manuals and technical catalogues.

When mass production is stable (right part of the picture), client orders procedure skip the development stage and orders are usually fulfilled in a reference basis, thus without any specifications concern.

The product specification settlement gathers all the inputs along the pathway of Figure 11, pathway leading successfully to a Final Product. Regardless herein the path is simplified and almost sequential, which may mistakenly seems a old fashioned work method, recent production paradigms (like Concurrent Engineering) gave more contribution in speedup specifications definition than innovative methodology in the specifications definition process. Enterprise and research efforts in DF*x* i.e. Design For Manufacture (DFM), Design For Assembly (DFA), etc. are examples of attempts towards achieving the goal of enabling the designers easily access the information related to various design aspects, including manufacturability, and that way establish satisfactory

[11] The shift from "mass" to "mass customization" production is typically illustrated with the Ford T cliché: "You could get any color you wanted as long as it was black", versus a today's paradigm where costumers can get any color they like.

product specifications in an adequate and timely manner. Hence, despite such methodologies effort to mitigate the lack of Manufacture information in upstream activities, still nowadays manufacturability information is not formally available [16]. Using informal

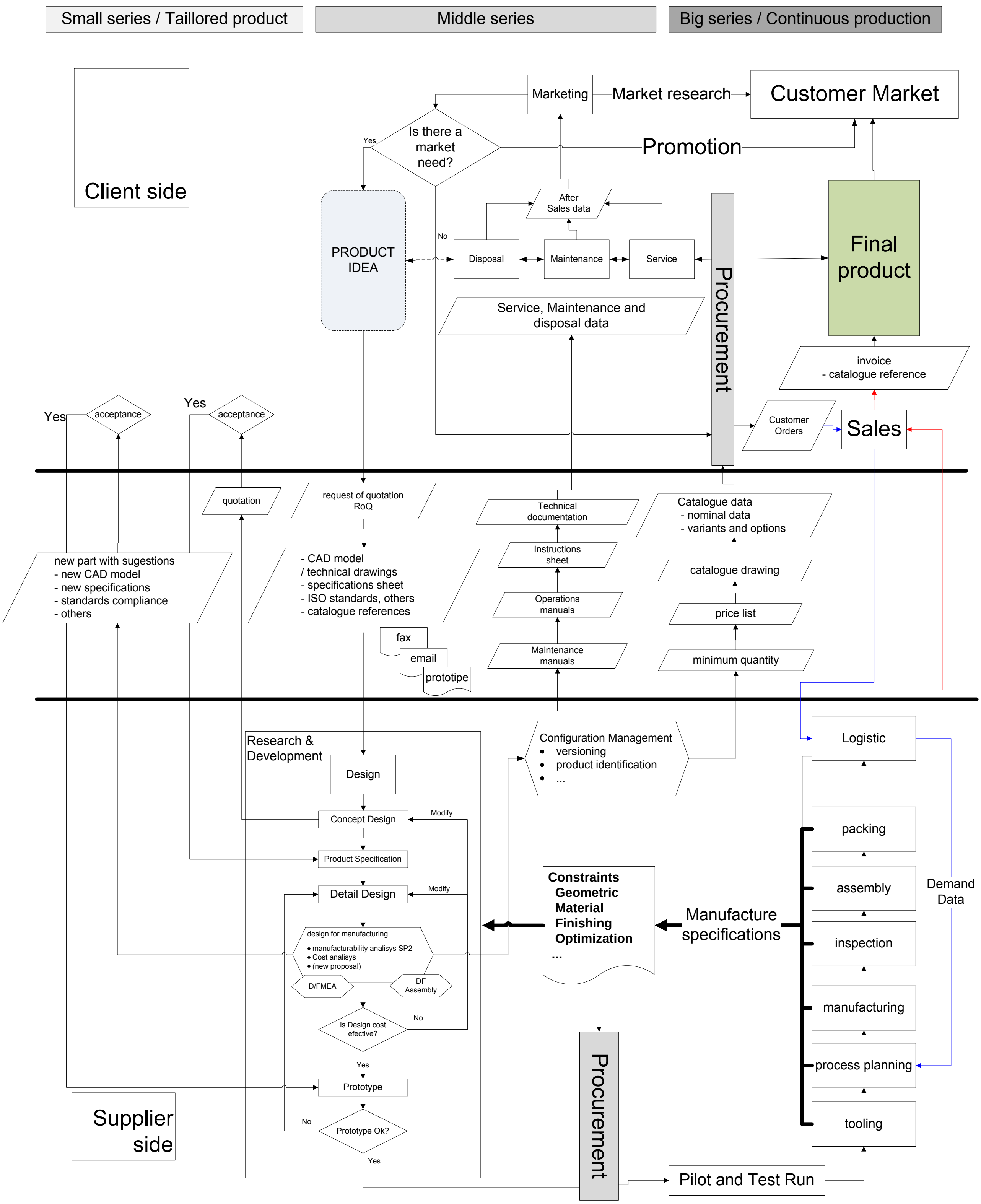


Figure 11 – Information exchange overview along PLC main phases.

manufacturability information in the early phases of the product development process can lead to e.g. untrustworthy feasibility study or unnecessary design iterations.

On the other hand, product specifications are essential as a basis for manufacturing planning: bill-of-materials and routings, among others, are mandatory specifications used for capacity and material planning. In Figure 11 Configuration Management is responsible to discriminate other specifications to the customer commonly used in the various life phases of the product (not only dimensioned drawings but also installation specifications, service specifications, other procurement requests).

Particularly in what Procurement respects, the illustrated phase frequently is not granted with a procedural product specifications list when Process Planning initiates supplier's auscultation. The content of specifications list are frequently incomplete, imprecise and with provisory values, which delays and difficult the subsequent evaluation process.

To ensure that such data and information can be reused, it is of great convenience that a structured, comprehensible and complete documentation of the product specifications be available.

Establishing product specifications

Costumer needs are generally expressed using their own language and lexicon. Expressions like "*The car's door should close smoothly*" or "*The motor needs to be powerful*" are examples. While such expressions are helpful in guiding a clear perception of the issues of interest to costumers, they provide little guidance how to design and engineer the product. It is simultaneously a benefit and a disadvantage: they simply leave too much margin for subjective interpretation, not pointing the final solution that satisfies the problem, but leaving margin enough to Design team to find the best compromise in "how" to solve the problem and optimize the solution.

Conceptual and Detailed Design then start by an iterative process usually establishing a set of specifications, which spell out in precise and measurable detail "what" values the Product must get conform. For example "*The engine shall have minimum nominal power of 75 kW*".

An expression with a variable and a metric is established. The variable is referred through a known label of a property and semantic or contextual meaning is usually associated. The metric part is expressed with a dimensional value associated with a physical unit. The expression uses a mathematical (engineered) relationship: equality, inequality or a range.

However, looking at the content of such document, it is a list of textual notes, reporting variables associated with metrics. Sometimes, a more structured list or spreadsheet is available. STEP technology somehow made a contribution in this field: AP 203 and AP214 included not only geometrical information but also relevant product information in a neutral computational way. Nevertheless, the interconnection between PLC phases still has serious shortcomings to achieve a complete integration and interoperable product specifications management.

As an example, in most SMEs, the validation of specifications which occurs in Inspection phase (see Figure 4) still relies in a human interpretation of product specifications set, regardless all computational skills of precedent activities.

Specifications refinement techniques

Within enterprise environment, the process of product specifications establishment could be classified in

- Manual;
- ICT supported ;
- ICT automated.

In the case of manually established specifications, more and more in disuse, some initial sketches and drawings are still found, which help engineering teams to conceptually define the product innovations that costumers aimed to see. Another example still available within some specific industries is the physical model or prototype, where specifications don’t assume an explicit condition. The human capacity to handle such specification activity is no longer sufficient, mainly due to product complexity increase and multidisciplinary teams more and more engaged in such activity.

Today’s information and communication technologies made available several tools to assist human tasks in evaluation of product specifications. PLM powerfully addresses the management of product data and information – both in content and in context: product and product families, project portfolios, documents and their associated content like Product requirements (functional, performance, quality, cost, physical factors, interoperability, production time, etc) or Technical publications (like Service Manuals, User Guides or Assembly Instructions) [7]. From CAD, to define precise geometric specifications, to CAE, to evaluate product behavior according to different disciplines, a lot of computational tools are used to aid the specifications process. Also data exchange between applications and interoperability of data formats was made under surveillance of research community, but concerns where more focused in wherever it was possible to import data from one application to another. A subsequent work of interpretation of the data exchanged comparison and matching is manual thus reasoning is made by humans.

Another important contribution to more and more product specifications refinement was Total Quality Management. TQM was responsible for the introduction of quite popular tools like QFD (quality function deployment) or FMEA (functional modal effects analysis) in automotive industry and in other industries. Again, all those tools and techniques still rely on human evaluation, and the re-use of the product specifications is guaranteed more through past experience of professional involvement than in the computational automation the task may have.

In what application of ICT to automating specifications establishment respects, it may be considered in many ways similar to the application of machines in physical production. Where physical production transforms an input of materials to an output of products, it could be aimed that the specification processes transforms an input of information to an output of specifications.

This level is characterized by the use of expert systems to make the reasoning, otherwise done by humans, and by integration of applications. Expert systems (in this context typically called configurators) make it possible to create a set of specifications automatically, based on a set of inputs. The ultimate automation is where there is an integration between expert systems and other applications, for instance between configurators, CAD/CAM, calculation programs, document generation and ERP. This makes it possible to operate entire processes without doing anything but entering desired product functionalities and structures.

In the scenario of tough economical competition, the expected increase on demand of product customization combined with new capabilities of information technology (like emergence of Semantic Web enhancements) will make specification process reengineering an interesting subject of interest.

In this sense, hopefully the present work makes a contribution in such aimed automated scenario in the near future to better manufacture specifications integration.

Chapter III

III. Trends in Manufacturing Enterprises modeling

With the evolution of the manufacturing paradigms, traditional environments inevitably converged to the new distributed and agile organizations. The evolution was felt not only at conceptual level, but also at the techniques and mechanisms associated to the re-engineering process like control and integration architectures for manufacturing systems [34].

This chapter presents a brief overview of the evolution of the manufacturing paradigms from traditional to the new and distributed organizational environments, as well as the requirements and problems associated to the innovative distributed and agile organizations. Also the techniques and mechanisms associated to the re-engineering process under enterprise modeling approaches are discussed, namely agents and multi-agent technology

III.1. Enterprise Modeling

The highly competitive global economy had lead companies to evolve to non-traditional formats or operation procedures. Within emergent and dynamic business fields, each company was confronted to fully understand the way it was operating and the need to align the organization structure with co-partners. Moreover, enterprise networks were proving to be a successful candidate paradigm, particularly enabling SMEs to increase their business trends. Several enterprises clearly assumed new formats like Extended Enterprises or Virtual/Agile Enterprises to guarantee product and resources control and the consequent integrated final service to costumers in a worldwide scenario. This is achieved forming and dissolving partnerships rapidly, assessing information and potential contributions of co-operating partners and integrating international business discourse intra and inter-enterprises.

Modeling the enterprise and its environment was a natural response to integrate organizational, economic, human and technical aspects and continuously maximize operations and their co-ordination through computational resources. This is a challenging task because several sub-model needed to be fused: organization model, resource model, economic model, functional model or information model, just to mention the more obvious process stages identifiable in any enterprise.

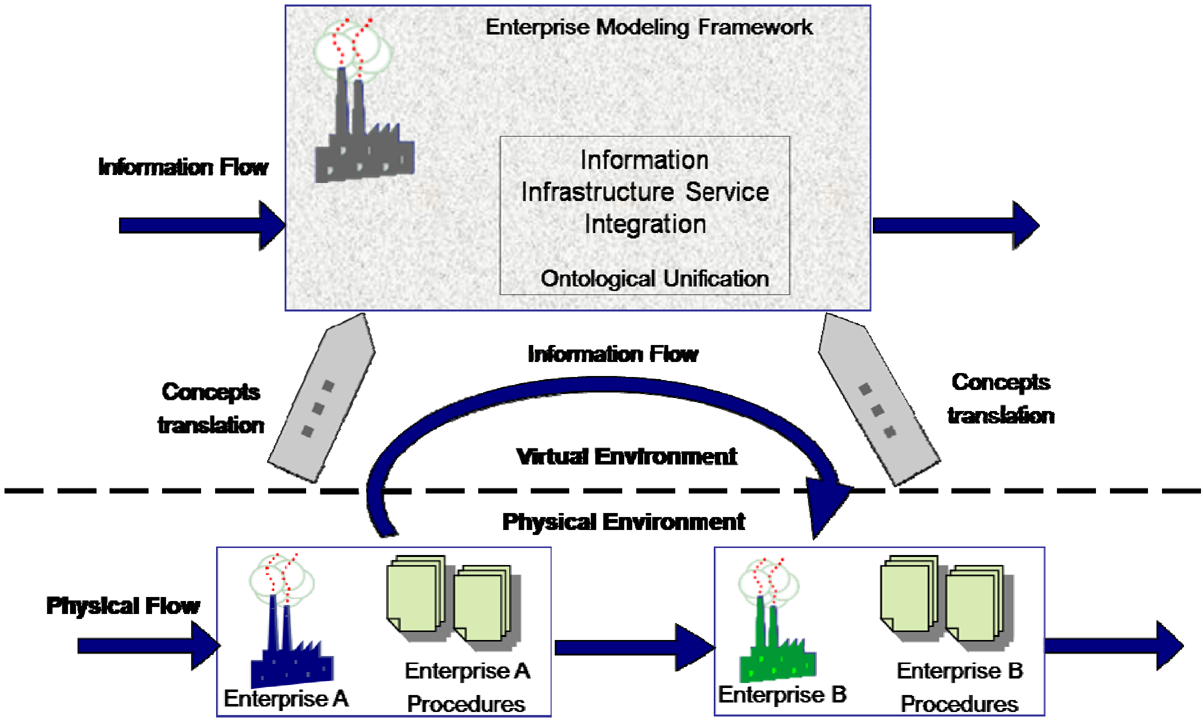


Figure 12 – The concept of Enterprise Modeling (adapted from [35])

Enterprise Modeling may be defined as the art of externalizing knowledge (which adds value to the enterprise or needs to be shared), i.e. to describe the things of the enterprise in terms of concerns function, behavior, information, resource, organization, economic or other aspects of a business entity. Enterprise Modeling may be used to represent the business entity to understand, thus enabling important processes like (re)engineer, evaluate, optimize and even survey and control business operations and performance. Figure 12 sketches conceptually the enterprise model framework linking the enterprise model to the real world systems.

The Figure 12 graphically distinguishes those two environments. In the bottom, the representation of enterprises' physical existence, where Enterprise A could represent the supplier of Enterprise B, and the upper part outlines the immaterial part of the environment. In today's information era, the information flow assumes a relevant role. In the Enterprise Modeling process, the translation of concepts and knowledge representation to the Enterprise Modeling Framework (belonging to virtual environment) could be established under reference architectures like CIMOSA or GERAM.

In effect, both academic and industrial community shared common objectives in the development of Enterprise Modeling approaches. Several main goals may be summarized in the following list:

- modeling should enable the understanding and explanation of the enterprise itself, thus constituting a diagnosis of existent disorder in terms of materials, information, and control flows;
- consequence of the precedent diagnosis, several steps are taken to restructure the business entity, thus improving its performance;
- once developed, the model should be used to experiment different scenarios, analyzing the past performance and anticipating the near future (batteries of tests, comparisons, predictions and projections could be carried out through such enterprise model, with minor impact in what consequences respect);
- also *what-if* scenarios using a model may help management to learn and to decide better options to the enterprise prosperity; usually occurring the Business Process Reengineering which leads enterprise to reorganize activities and efforts;
- the modeling process usually serves to build a common consensus on how enterprise operations work (or should work) and to turn the organization structure prepared to face business changes;

- the model should enable large scale systems integration, promoting interoperability of main application systems like MES, ERP, PDM or APS and conformity to standards like ISO 9000 or ISO 14000 (or hopefully alignment);
- modeling contributes to Management decision transparency due to externalization/internalization of its activity; internal access to management rules and principles

A primary profit of the business modeling process is commonly the re-engineering, or at least, the clearness and discipline introduced in enterprise information flows. The information flow internal and external to enterprises is long since an important resource, and his mastering is a key factor to enterprise success. Figure 12 remarks a particular aspect of information flow. Enterprise Modeling aims the integration of Information Infrastructure Services with other models, which require harmonization characteristics. The use of a common terminology and more recently semantic and ontological requisites in such information flow are main requisites to effective transfer of information across and beyond the organization.

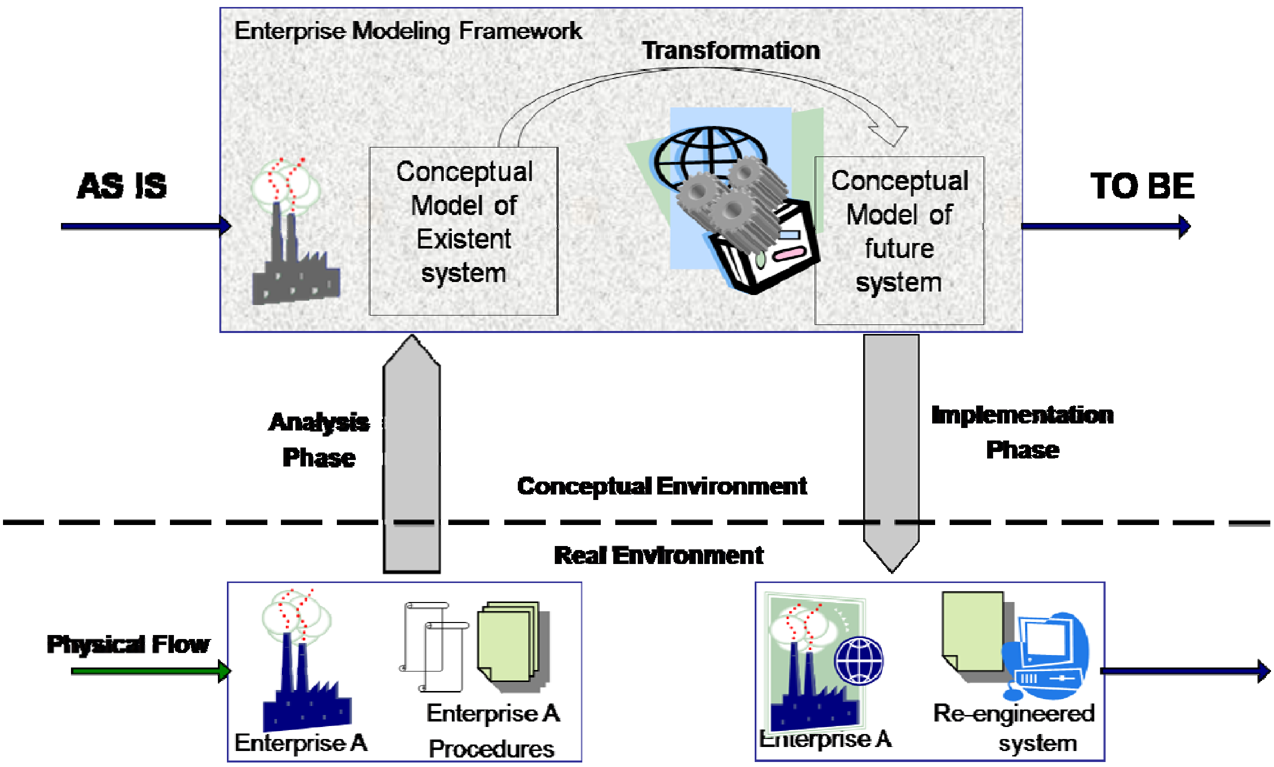


Figure 13 – The Enterprise Modeling activity process.

The picture of Figure 13 represents how to explore the potentialities of enterprise modeling. AS-IS and TO-BE scenarios are quite useful in virtual environment (limiting as a way to organize and re-use implicit and explicitly knowledge that has been modeled and expressed.

The model of the operational processes provides the explicit knowledge on information needs during the enterprise operation. Such model helpfully identifies the operational tasks, their required information supply and removal needs as well as the point in time of information transactions. Therefore, the modeling process has to be guided and supported by a reference architecture, respective methodology and information technologies based tools.

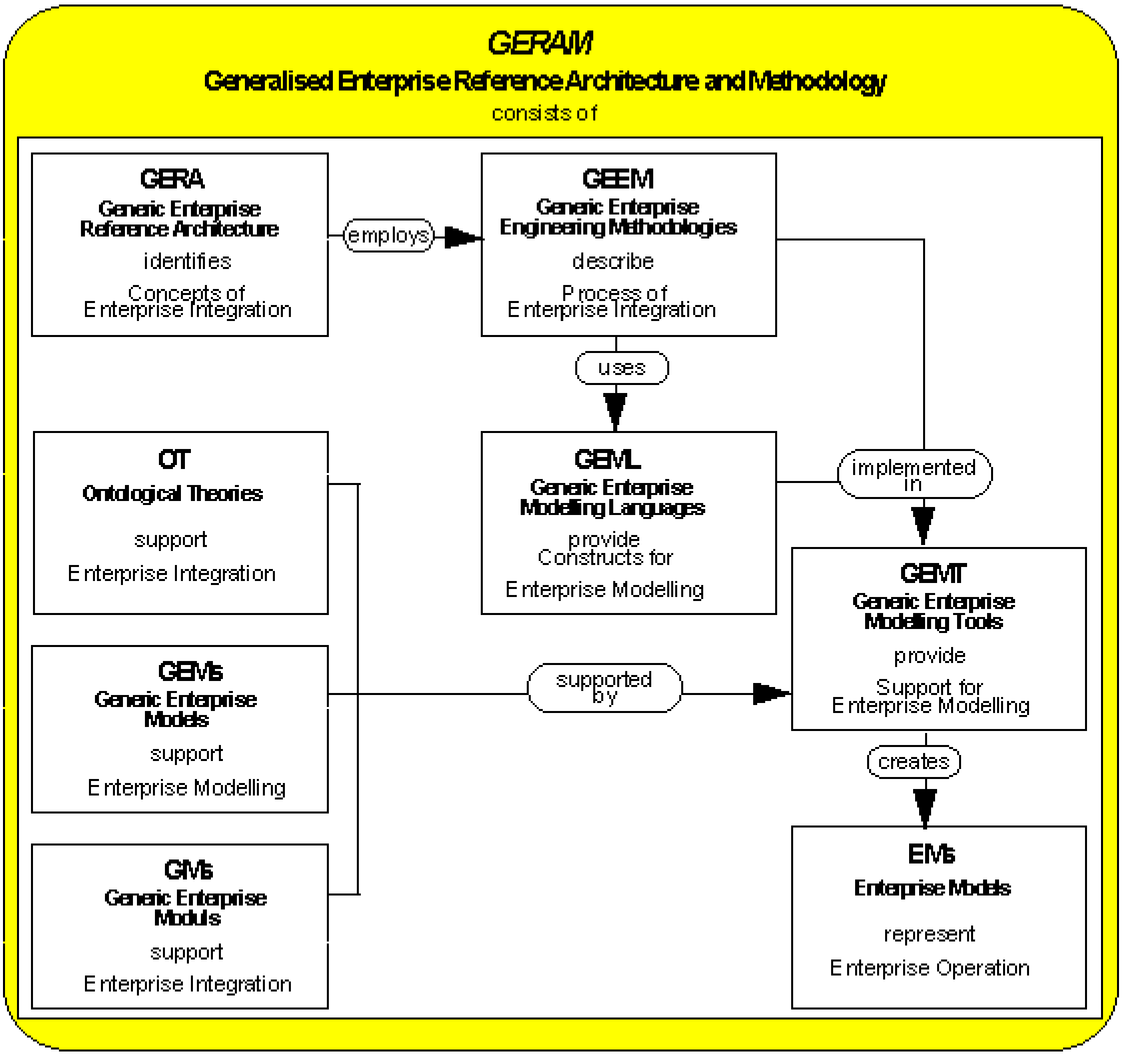


Figure 14 – Example of reference architecture for Enterprise Modeling methodology (adapted from GERAM)

Several reference architectures for Enterprise Modeling were proposed to bring the concept to daylight. Among them, some become popular after successful applications: CIMOSA (Computer Integrated Manufacturing Open System Architecture), PERA (Purdue Enterprise Reference Architecture), GRAI (Graphs with Results and Activities Interrelated), IEM (Integrated Enterprise Modeling), ARIS (Architecture for Information System), GERAM

(Generalized Enterprise Reference Architecture and Methodology). The Figure 14 provides the example of GERAM framework for Enterprise Modeling.

Numerous computational tools may be identified that give support to enterprise modeling:

• ARIS ToolSet (IDS Scheer)

• FirstSTEP (Interfacing Technologies)

• Metis (NCR)

• PACE (IBE Simulation Engineering)

• MooGo/IEM (IPK Berlin)

• CimTool (RGCP)

• GraiTools 1.0 (GraiSoft)

The list is not exhaustive nor has serialization pretension.

The main international standardization bodies like CEN and ISO also made strong efforts to develop standards compliant with enterprise modeling infrastructure and services. Relevant standardization like STEP/EXPRESS, EDI or SGML are visible faces and still in progress. Parallel international initiatives like IFAC/IFIP Task Force, Workflow Management Coalition (WfMC), CALS, QCIM in Germany, among others also seek harmonization in particular areas of enterprise modeling.

Moreover, European ESPRIT project, American NIIIP or the world-wide IMS organization endowed several projects addressing enterprise integration as a subset of enterprise modeling [35]. One of the problems of enterprise models is its consistency and integration. In fact, integration issues have been discussed since the early days of computers in industry in general and in the manufacturing industry in particular [7].

Thus, despite the extension of listed advantages of enterprise modeling, existent information technology does not yet assures a fully integrated solution for quotidian use by operational staff. In addition, application of existing enterprise modeling technologies has been hampered by a lack of business justification, excessive conflicting solutions and terminology and by an insufficient understanding of the technology by the end-user community [25]. The small-to-medium-sized enterprises (SMEs) are once again confronted with serious barriers to the use and deployment of pragmatic and valuable methods and tools in this domain, in contrast with their big sized sisters.

Moreover, a huge number of computational tools claiming to support Enterprise Modeling still need further refinements to fully contribute at a desired level. Envision of future scenarios in what change management respects and effective help to decision support of end users are improvements not yet achieved.

III.2. Modeling of Manufacturing Systems

Manufacturing Systems

Manufacture enterprises are not exception in the scenario of enterprise modeling. The increase of competitiveness of current days is expressed in more productivity, more quality, more agility, more flexibility and better adaptation of manufacture enterprises. These are challenging objectives in order to enterprises stay in the business, but solution is found by cooperation between the enterprises. This was an opportunity for Small and Medium Enterprises (SME) to get access to new markets in the global economy, participating in supply chains and forming virtual enterprises and e-alliances to fulfill specific customer demands.

Classification of Manufacturing Systems

Depending on the type of enterprise, manufacturing or process industries, the completeness of modeling process is stratified in different levels. The core business of production enterprises is the materialization of products. But manufacturing industries reveal far different challenges in comparison with process industries. The first pose more dynamic issues in what knowledge capture respects (since each discrete final product may represent a new iteration process), and the next possess more stable process variables and respective control, facilitating knowledge capture. Mastering the modeling process in the process industries may be easily reproducible, which is not the case in the manufacturing enterprises and their manufacturing systems. These will be the focus of the present work.

Typically, a classification of Manufacturing Systems considers the type, layout or volume of production.

Taking into consideration the production orders to establish a division, in terms of type of manufacture one could have:

- make-to-stock, where the production is done for the stock, based in forecast orders, such as in the high volume textile and shoe industry or standard mechanical components;
- assembly-to-order, where final products are only assembled after receiving a customer order, such as the automobile industry or domestic electrical stuff;
- make-to-order, where the production of the product starts after receiving a customer order, such as in the case of production of machine tools;
- engineer-to-order, which is an extension of make-to-order type, where one-of-a-kind products are designed and manufactured according to the customer specifications, such as in the space electronics.

Another possible classification of a manufacturing system considers the production volume, which according to [36] results in the following categories: job shop, batch and mass production.

The job shop production is characterized by the production of small quantities, often one-of-a-kind, of a great variety of products. Typically, the equipment used in this production type must be flexible and general purpose to support the great variety of products. As examples of job shop production, it is possible to mention the production of machine tools, molds and aircrafts.

The batch production involves the production of lots of medium sized quantities of the same product, which has a regular but not so high demand. The equipment used in batch production is general purpose specially designed for higher production rates, for example a tool machine equipped with special fixtures designed to increase the machine production rate. As examples of items produced using batch production it is possible to find electronics equipments and furniture manufacturing.

The mass production is related to the specialized production of one or a small number of products, each one with high production rates. The equipment and the factory plant used in mass production are completely dedicated to the production of a particular product. Examples of items produced using this type of production are screws and light bulbs.

With minor interest to the present work, yet another possible classification of a manufacturing system is considering the physical plant layout. The author considers that this division does not bring significant impact in the current work approach (since the disposition of the physical facilities raises relevant issues in terms of production times and logistic concerns but not in terms of final product characteristics).

The evolution of manufacturing systems: which input to modeling process?

Despite the precedent considerations about classification, the current trends of manufacturing systems seem to be the mass customization where every single item of a product needs to reflect client's specificities.

For several years, the mass manufacturing concept was a productive mean to have massive production numbers, to downsize costs and to quickly respond to avid and brand new markets. However, the rigidity in variations of the kind of product and waiting times to client (that changes of manufacturing chains implied) become serious obstacles and clear causes on markets slow down. Only some products survived to such manufacturing paradigm. Extending the concept of just-in-time, the Lean manufacturing successfully shortened the timeline between costumer order and shipment, managing to have the right material at right place at right time through the reduction of stocks and general production waste. Similar to Lean manufacturing, the concept of Agile Manufacturing was introduced providing the manufacturing environment with the ability to adapt quickly and profitably to continuous and unexpected changes. Not only such paradigm enabled diversity of offer, but it also fulfilled the requirements of the globalized market for reduction of prices, better product quality, minimum delivery time, that is, more competitiveness.

Mass customization is becoming so widespread and popular that seems to be the production paradigm in future, and Internet era has provided a great contribution in that direction. However, a lot of work must be done to adequately get an environment with traditional manufacturing systems modeled and adapted to dynamic and challenging environments.

Looking to some evolution in this field may help to understand contributions that are still lacking of an efficient and fully operational enterprise model for modern manufacturing systems.

Flexible Manufacturing

Mass customization introduced the problem of a variable and customized demand. As an alternative to smoothly answer to fluctuations of demand producing an excessive quantity and stock, Manufacture enterprises preferred the increase of flexibility of manufacturing systems, dealing with the production volume and variety. This preferred solution was aided through automation technologies that industry was being provided and still nowadays has available, such programmable logic controllers, industrial robots, numerical control machines and succeeding computer numerical control, automatic guided vehicle systems and automatic storage systems. A natural consequence of the flexibility increase of such a dense system was an increase in complexity of its control.

Flexible Manufacturing System concept was on the origin of a set of workstations interconnected through a material handling system being controlled by a centralized computational system [36]. Once operational such hardware part of the flexible manufacturing system, the optimization of the combination of operations involved in a one-of-a-kind product was an accessible task that production orders solved. It was remaining the optimization of the manufacturing control as a whole computational task. It could be said that this virtualization were a rough envision of current enterprise modeling concept.

Computer Integrated Manufacturing

The automation of partial stages of manufacture systems solved successfully some lead times and optimization issues. Nevertheless, done in a standalone way, information redundancies originated automation islands and become a key obstacle to the aimed integration.

Only through the Computer Integrated Manufacturing concept, consisting in the integration of enterprises activities through the use of emergent information technologies, further progress was made. Interconnection of databases and networks, enabling the share of the production data, became reality and evident advantages resulted from the integration of several computer-aided technologies supporting manufacturing systems such as Computer Aided Design (CAD), Computer Aided Engineering (CAE), Computer Aided Process Planning (CAPP) and Computer Aided Manufacturing (CAM).

Nevertheless, still ongoing efforts aiming the CIM paradigm implementation, though technological heterogeneity, in parallel with social and economical problems [37] strongly difficult a visionary integrated manufacture. The problems arise at several levels: proprietary protocols in data communication between each equipment and technology; need to raise employee’s skills to properly handle the system, difficulties in expansion and process reconfiguration for new products manufacture and consequent lost of flexibility, among many others. These integration issues only deal with the engineering and manufacture activities inside the same organization. Much more arise when such integration attempt is extended to all activities related with manufacturing activity. The integration of the enterprise systems with suppliers and customers systems, giving birth of

Supply Chain Management concept, exposed even more issues namely in interoperability and compatibility fields.

In plus, in most of the cases, budgets involved in successful integration were incompatible with profits and return of investment of such manufacturing systems.

Distributed Manufacturing

Recent manufacturing industry is characterized by active enterprises operating in autonomous production facilities and cooperating among partners both locally or worldwide. The previous referred concept of Supply Chain Management identifies itself more with this organization manufacture system, recognized as Distributed Manufacturing. By definition, a *distributed manufacturing system is a production system that is geographically distributed, acting in a cooperative way in order to work as a whole* [34].

Distributed Manufacturing Systems deal with the management of materials, information and financial flows in a network, consisting of suppliers, manufacturers, distributors and customers, and aims at having the right product in the right place, at the right price, at the right time, and in the right conditions.

A complex and continuous control of the manufacture flow is required to keep consistency, hence the complexity of the system demands an extreme and careful division of the responsibilities among each involved enterprise. Moreover, profiting with different skills and resources of the associated enterprises, the whole offers more than the sum of the parts. This concept was named Virtual Enterprise, which paradigm is defined as a temporary alliance of enterprises that come together to share skills and resources in order to better respond to business opportunities and whose co-operation is supported by computer networks [38].

The manufacturing enterprises can no longer prosper with stand-alone procedures or within rigid supply chain, but need to constantly reconsider how they are organized. Although concepts are not yet well distanced and well-defined, other forms of virtual enterprise organizations emerged, such as the Extended Enterprise, e-Alliances or Smart Organization, with several cases namely in automobile industry [25].

Moreover, today's products are offered having several version models, and each version model can be highly tailored in order to fulfill the customers' requirements. According to early classification of manufacturing systems only production under job shop type or in some cases the one-of-a-kind was able to respond to such tailored requests. Within agile adaptation to the environment changes, customers started perceiving manufacturing as a mass customization, and the companies were obliged to use new technologies and new manufacturing concepts, and to combine them, in their different activities to avoid the risk of becoming less competitive or even obsolete [34].

III.3. Agents

In an abstract level, the term *process* can in general be defined as a change in the properties of an entity, including geometry, hardness, state, information content (formally data) and so on. To produce any change in property, three essential instruments must be available: *material*, *energy* and *information*. Depending on the main purpose of the process, it can be classified as a material process, an energy process or an information process [39].

Figure 15 – Main entities present in a *process* necessary to Product properties modification.

Process flow examples:

material – like raw-material, machinery or tools;

energy – like electric power and human labor;

information – both tangible like databases or results and more intangible like experience and knowledge.

The modeling of a manufacturing industry needs to consider a platform that comprises those three main flows.

In the past, cost's structures of products had prominent factors like cost's machinery, raw material and human labor. Nowadays, information plays a decisive role in comparison with material and energy in the production process and knowledge is a capital factor in virtual environments of manufacturing systems.

The agent technology was introduced as the suitable solution to support enterprise modeling of manufacturing systems in what information flow concerns. According to

Parunak [40], the manufacturing applications require adequacy to characteristics like modularity, decentralization, changeability, ill-structured and complex problems handling, just to cite the most relevant [40]. In spite of a non consensual definition of agent, here it is considered as

An autonomous component, that represents physical or logical objects in the system, capable to act in order to achieve its goals, and being able to interact with other agents, when it doesn't possess knowledge and skills to reach alone its objectives [41].

During concepts translation phase of enterprise modeling (see Figure 12), an agent can represent not only physical resources like machine tools or components transportation systems but also logical objects such as schedulers and orders. Moreover, since the domain is manufacturing, an agent may be seen as a software entity that represents the resultant operations either from physical devices or from human tasks and functions [42].

A high-level overview of the manufacturing systems comprises specialized functions, knowledge, skills and operations interacting (both internally and externally to the enterprise and at different layers) in a cooperative manner to achieve shared and common goals. Therefore, the control of such a complex and heterogeneous systems requires a set of autonomous components in order to rearrange organization into cooperative, dynamic and distributed structures in a simple and non-specialized intervention.

III.3.1. Multi-agents in the manufacturing domain

As described before, the global market is more and more distributed and settled with heterogeneous environments and assisted by heterogeneous hardware and software applications. The proliferation of computational resources amid manufacturing systems, reaching different levels and skills, may be in the origin of enterprise modeling through a layered approach [43]. In the bottom of such stratification is the integration of/with physical resources, like CNC centers or PLC equipment, and as one goes up in the layer structure, functions and responsibilities are modeled entities interacting virtually.

An approach committed to the agile manufacturing issues, with particular success in operation within distributed organizational environments, is the multi-agent technology deployment [37]. Figure 16 exemplifies different agents' deployment in a production unit model.

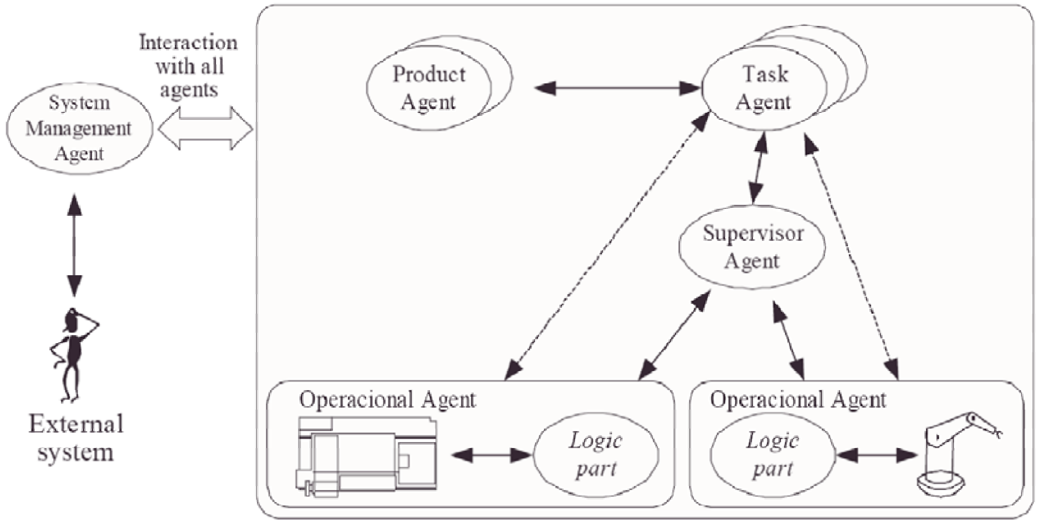


Figure 16 – Example of Agents considered in a modeled production unit (from [26]).

The concept of multi-agent has its origins in the Distributed Artificial Intelligence computational field. According to Ferber 1999 it may be seen as a *set of nodes*, designated by agents. In turn, agent is conceived as a component of software and/or hardware capable of acting and making decisions in order to accomplish tasks.

Similar to distributed artificial intelligence, in distributed manufacturing environment the multi-agent has proven to be a suitable technology to proportionate significant support to Enterprise Modeling. In terms of benefits, multi-agent technology proportionates [42]:

- autonomy: it is supposed that an agent operates without further intervention of external entities, having control of its own behavior;
- cooperation: the net of agents has interaction in order to achieve a common goal;
- reactivity and/or pro-activity: agents perceive changes in their environment and respond accordingly and in an adaptive performance;
- re-organization: in de-centralized environments, agents may be re-organized into different organizational structures [37].

But agent technology has its drawbacks, the most relevant being the interoperability issues. In fact, dissemination of the technology and widespread implementations through the internet put on evidence that lack of systematization and normative procedures when developing agents.

Nevertheless, several successful research projects in different application domain revealed the potentialities of agent technology, and particularly in the manufacturing systems field. Barata and Camarinha-Matos [44] focused on the shop floor re-engineering, using agents to represent the physical components which are aggregated into consortia regulated by contracts, to agile the shop floor life cycle.

In the area of enterprise integration, CIIMPLEX [25] uses agents to represent the communication services, such as name server, facilitator and gateway, aiming the integration of planning and execution in manufacturing systems, in real scenarios, involving legacy systems (execution system and scheduler system).

Rabelo and Camarinha-Matos [45] under HOLOS architecture, used multi-agent based dynamic scheduling. The flexibility of the architecture both from the organizational and from the control point of views is supported by negotiation techniques and the dynamic formation of consortia of manufacturing resources. One of the important aspects of Rabelo *et al* work is the tandem agent architecture to support integration with legacy systems. This work was later extended to distributed multi-site manufacturing systems and virtual enterprises in the framework of the MASSYVE project [46].

Pithon [25] used a multi-agent platform under the BR_VEARM architecture environment (defined by Putnik [26]) to validate a new organizational specification structure for concurrent engineering in Virtual enterprises.

Agent-based approaches are extensively applied with success in many different areas. Besides manufacturing field, electronic commerce, e-business, air traffic control, process control and telecommunications are examples with proven success. In effect, agents are proving to make great contributions in the e-business and in particular in electronic commerce (e-commerce). The advent of internet era, enabling enterprises to do business in a global scenario, demanded new tools and approaches to successfully improve trades using the electronic market.

III.4. Languages supporting Enterprise Modeling

The Agent-based technology has an additional advantage in comparison to complete integrated solutions: required resources in terms of software and human skills to develop agents are more accessible and simpler than in a centralized solution. This makes development easier, with less exposure to shortcomings and future maintenance issues.

Sometimes some confusion may arise between the agent-based and object-oriented approaches. A major reason relies on the use of object-oriented programming languages, such as Java and C++, to implement agents. While objects in Objected Oriented (OO) languages are passive in their behavior, thus only reacting to external stimulus, like a message or invocation of a method, agents are provided with the ability to decide requested services by other agent. This encapsulates the other major difference: objects have a set of attributes and a set of services and methods, but do not possess the capacity to choice whether they will execute or not services requested by other objects. In its turn, based on embed knowledge and skills, agents decide the service execution, or even evaluate if data received or to send are simple information requests.

A particular and suitable case of object oriented language, helpful in Enterprise Modeling is Unified Modelling Language (UML). Among other Enterprise Modeling techniques and alternative methods, UML is a specialized language well suited to Enterprise Modeling requisites.

Unified Modelling Language was developed to assist software system developers to specify, visualize and document models of software system, like their design and composition [47]. The UML innovative technique still lasts today due to the small number of standard diagram types that are used – about thirteen – and corresponding easy analysis and overview of the modeled system. Additionally, it could be used UML to model any type of application, running on any type and combination of hardware, operating system, programming language, and network. In what non-expert experience respects, UML enables an user friendly and quick develop environment of a software model, or any other non-software model like those required in business or engineering fields.

UML presents several advantages in what modeling concerns: it is flexible enough to enable distributed applications modeling integrating middleware on the market; it was built upon fundamental Object Oriented concepts including class and operation, becoming adequate for object-oriented languages and environments such as C++ or Java.

Several UML Profiles were tailored to specific areas - some for business modeling; others for particular technologies like Transactional modeling, Real-time, and Fault-Tolerant systems. Standard Profiles are available online and the list includes, as bellow [47]:

- UML Profile for Software Radio)
- UML Profile for CORBA® and CORBA® Component Model
- UML Profile for Enterprise Application Integration (EAI)
- UML Profile for Enterprise Distributed Object Computing (EDOC)

- UML Profile for Modeling QoS and Fault Tolerance Characteristics and Mechanisms
- UML Profile for Schedulability, Performance and Time
- UML Profile for System on a Chip (SoC)
- UML Profile for Systems Engineering (SysML)
- UML Testing Profile

The profile specially developed for Systems Engineering has been thoroughly explored by Georgia Tech research community, which developments and achievements are remarkable in manufacturing knowledge management systems.

III.5. The e-procurement

Initiatives to support electronic data exchanges in business processes between different companies are not recent, although electronic commerce suffered a relevant progress in the outcome of the internet era. Nowadays, electronic commerce is an unavoidable reality for company's competitiveness, both for suppliers and costumers. The initial business-to-business solutions have the exclusive purpose to profitably exchange financial information, after successful computational communication network settlement. The resultant advantages easily incentivized the exchange of product's technical details in complement to financial ones.

Enabling enterprises to do business in electronic media demanded new tools and approaches to successfully profit with electronic market. The Procurement activity necessarily re-adapted itself to such reality, and catalogues in electronic format are now available online in every company's website. The rapid evolution of electronic data exchange formats and struggle against time to answer market's demand did not contribute to a methodological solution in terms of the emergence of a formal catalogue structure. In today's scenario, every electronic format is used to electronically share information about company's products.

At first, in order to exchange business transactions, costumer and supplier needed to agree on a common structure (a protocol for transmitting the content and a language for describing the content). A number of standards arose for this purpose. One of them was the United Nations initiative Electronic Data Interchange for Administration, Commerce, and Transport (EDIFACT)[12].

Currently, business transactions that are supported by web browsers have more integration capabilities since HTML based web pages were migrating to XML format, with standardized information structure [48]. XML emerged as robust format for data exchange for its openness and explicitly level of data as well. Even STEP community developed a Part 28 module to express Product Data models in XML. Regardless the progress impact of XML providing a serialized standard syntax for defining the structure of data, semantic description of business processes and exchanged products is missing.

A great contributor to data structure of Procurement activities was PLib initiative; the available PLib Standard (officially the ISO 13584 Standard series "Parts Library") has defined a model and an exchange format for digital libraries of technical components.

PLib initiative discovered semantic requisites need when components' technical information exchange (thus among a community, like automotive or aeronautics) still presented interoperability issues, due to different meanings of the same variable or similar concepts with unclear definition and meaning.

More recently, PLIB introduced recently OntoML as part of ISO 13584 specification to use the Extensible Markup Language (XML) and the XML Schema specification for representing data according to the ISO 13584 data model.

[12] Nevertheless it was exclusively focused in financial entities and issues were arising when some nuances at syntax allowed changes did not contribute to the aimed interoperability.

Chapter IV

IV. Semantic support to an integrated PLC

Today's product information models of a mechanical component can be considered a complex data model, due to its multi-to-multi relationships among properties list (which extension is becoming also considerable) and with different data level, addressing several phases of product life-cycles. One particular problem emerges when members of a team have different backgrounds (engineers that design the product, manufacturing people that fabricate it, or more and more service people that perform maintenance and repair) and data becomes spread and not modeled under the same structure, compromising seamless integration of planning and process levels.

It is not unusual that enterprises often match this scenario, because the data itself is highly distributed between supplier and costumer and there is no global database culture, despite PLM systems aid an evolution in this field [20]. Therefore, ontologies are foreseen as a promising research field addressing these issues.

IV.1. Data, information and knowledge exchange

Considering the sentential examples of Product Specification of Chapter III, several notions about data, information and knowledge needed to be clarified in the remaining of the thesis.

IV.1.1. Data exchange

The conceptual requirement "The engine must be powerful" may be considered a sentence without computational interpretation purposes. It may result of an early design specification after client speech. In its turn, the engineered sentence "Engine shall have minimum nominal power of 75 kW" corresponds to a human description that is more easily translated to a mathematical formula, thus machine readable and computable.

In both cases information and knowledge is supposedly carried out with the data exchanged[13]. In the traditional development environment, raw data often has little meaning and is of limited use, obliging salesmen and engineers to additional communication, to clarify and to guarantee a correct and same interpretation (thus information and by consequence knowledge). All along, modern PLM systems use powerful resources to model the product and the process as well as related information. Little could be expected in semantic field, yet it is possible to maintain a wide-ranging and powerful database fulfilled with product or process data.

Typically in manufacture domain, the manufacture data that is maintained on a product database may include – among many others – CAD files with drawings, CAM files and some information on the manufacturing process. Tiny information will be maintained on initial requirements that implied established manufacture specifications, the process that was used to develop the data, or the choices that were made to get this data. Most of those underlying pieces of information will be unavailable or inexistent the next time that similar activities are carried out, thus to fully benefit from past experience will no longer be possible.

[13] According to several ISO committees, and in particular TC184/SC4 [32]:

- **data**: a representation of information in a formal manner suitable for communication, interpretation, or processing by human beings or computers.
- **data exchange**: the storing, accessing, transferring and archiving of data.
- **information**: facts, concepts, or instructions.
- **product**: a thing or substance produced by a natural or artificial process.
- **product data**: a representation of information about a product in a formal manner suitable for communication, interpretation, or processing by human beings or computers.
- **product information model**: an information model which provides an abstract description of facts, concepts, and instructions about a product.

Moreover, intentionally and for pragmatic reasons, a lot of information is discarded during the design cycle. Within the same company or a particular PLC phase, like design cycle, this may not be a major problem because the people involved can remember what has been discarded. It becomes a problem, however, when the data is transferred to other users in another part of the PLC phase, like manufacturing, where responsible will not be aware of some design intent, and thus not capable to evaluate the best manufacture options. It is also a problem when an attempt is made to reuse information at a later date. By this time, the original users of the data will have forgotten what they need to rediscover. Time will be lost as they try to find missing data, or develop new data to replace missing information.

Taking the example of CAD and CAM files (considering both raw data): they do not provide complete specifications (neither are supposed to) of the product, the various choices and activities that took place during design cycle used to develop the manufacture specifications they carry, or the process planning restrictions of a specific set of existent machines in a specific company shop floor.

The company's existing data is valuable, but to unlock its value, users have to be able to use it and know its meaning. Unless the full meaning of data is available, availability will be worthless in the future.

Current systems and applications generate and record a lot of product data and such support from data management systems is needed so users can have access to data in order to understand the full meaning of the data when required. If data has to be re--created or renovated manually, for sure errors will be introduced and time will be wasted.

IV.1.2. Information exchange

The human capacity of data interpretation is the responsible for the success of information share among interlocutors, after physical data exchange. In computational community, more than the syntax or grammar of the language carrying the data it was the lack of semantic property that was missing for the systems to be able to confer meaning to data.

The common way to specify meaning is by giving a definition, thus contextualizing the data. The international standard ISO 1087 declares ‘definition’ as a “*verbal description of a concept, permitting its differentiation from other concepts within a system of concepts*”. A definition can be given either in informal language or, in the case of the

Artificial Intelligence community, in a formal language that has a formal semantics. The semantic web is about formalizing definitions.

Informal semantics is about creating meanings that people can understand, whereas the formal semantic is about creating meanings machine-processable. It results then that formal semantics, mostly for machines, enables automated reasoning and consistency checking of the data. Thus, and if definition are related to terms, also terminology becomes an important notion.

This is another barrier consistently encountered when facilitating the sharing of information within Product Life Cycle: teams have distinct terminology or inconsistent lexicons [34] [49]. Unfortunately, the terminology found in the literature of classic disciplines is not at all standardized. Therefore, if a piece of terminology is adopted within an engineering design community, it only means that a substantial number of workbooks and original documents adopted it and use it with the same sense.

As an example of vagueness in terminology, Silva et al [19] identified among catalogues of the same product several examples of terms with traverse connotation: the same terms referring different product entities, different terms meaning the same product entity and cases of standardized terms with slight nomenclature variations to denote the same product.

Design and Manufacture use a widespread terminology and the large assortment of techniques described in the previous chapters used during design and manufacture, can easily lead to confusion and poorness of data sharing. The fact that interoperability issues are a reality among enterprise or manufacture communities, just to mention those, is a proof of the missing lexicon.

The creation of formal, widely accepted definitions of manufacturing lexicons would prevent many miscommunications among the engineering community, providing a much needed consensual understanding of the terminology. In a certain sense, this was started by standardization bodies, but due to its optional nature still some professionals and companies are reluctant to their adoption. Surely such a large undertaking will require the consent and recognition of the community, but most probably a consensus will be achieved only in a specific PLC phase community.

IV.1.3. Knowledge exchange

Chapter III described several methodologies (e.g.: Concurrent Engineering) introduced along Product Life Cycle to go beyond information exchange shortcomings of the involved teams. Despite the main focus was on time and cost reduction, the aim was to accelerate iterative process of design and manufacture teams. They were in a certain point of view as a shortcut to solve in the immediate the problem of information exchange among involved multidisciplinary teams.

However, the abstract knowledge (expertise or past experience as it is denoted in industrial environment) was an ambitious objective, that existing technology or methodologies did not address in a valuable manner. The term abstract knowledge may include the rationale behind an engineer’s decisions in the creation of a product model or its instance, such as why a particular shape was chosen, why a constraint exists, any existing model limitations, or any model justifications and objectives.

Addressing the need to capture the abstract knowledge used by engineers in the development of an engineering product model is expected in the next generation of software and PLM systems. Although some sort of knowledge, such as parameter values, may be accomplished by current software programs, through either third-party programs or interoperability between programs, abstract knowledge remains unaddressed. With this knowledge readily available, it will become easier to understand a product model as well as decisions made during its creation.

In current design systems, a modification to a design model may cause the subsequent models associated with the design to lose their applicability, requiring them to be replaced by a new model. Without knowing and understanding the knowledge used to validate the applicability of a model, modifications on a current model may be done incorrectly. By capturing abstract knowledge, the engineer will be provided with a better understanding of the model, knowing what modifications may breach its limitations. If the limitations of a model are exceeded, possessing abstract knowledge may allow an engineer (or a software system) to simplify an existing model (or identify if a new model is needed), thereby expanding its application to comply with its new domain. Moreover, improvements in knowledge systems may also be used to help "renovate" existing or even incomplete data.

As the necessity for capturing and sharing information has increased in accordance with the growth of computers and storage capacity, new methods are being developed to manage knowledge. A widely accepted formal method for the capturing of this knowledge is now being addressed by multiple research communities, in particular those related to AI and semantic web. In the following sections, a method for knowledge modeling using ontologies will be introduced, highlighting the advantages of efficiently capturing abstract knowledge of PLC Manufacture phase.

IV.2. Ontologies as a specification mechanism

Recently, research community, recognizing that it is needed a proficient way to computationally capture definitions of terms, initiated the development of several initiatives both in Manufacturing and Enterprise domains. The most visible step may be the Semantic Web initiative, though sharing common difficulties with other research communities.

Upon searching for the best conceivable way to capture definitions of terms, the engineering community adopted the suggestion from the philosophy community (which, after all, has been pondering the question for a number of centuries). It has been foreseen for that community the use of first-order logic as a suitable and appropriate answer, which has the ability to reason over sets of definitions and to prove properties of these sets.

One such property, for example, is *consistency*, which is tedious and prone to error using human or traditional information-modeling techniques to prove it. Using logic, it becomes straightforward to ensure the consistency of assertions for large sets of definitions by using automated theorem provers. Yet another property is *inference*, a mechanism that the use of logic certainly will enable and progress. If the data is becoming completely digital, computers should be enabled to somehow find the relationship among such data. One such empowerment is after all the essence that Artificial Intelligence community has being addressing for so long.

Two examples may be used to illustrate how helpful logic based knowledge representation formalisms would be to the mechanical engineering community.

Manufacture specifications are usually included in dimensional and additional notes to a product model (as reflected in technical drawings). Product models, such as STEP product data models, are computationally shared among engineers containing precise geometric and topological product information. Nevertheless, browsing a component's database respecting a simple specification *"The nominal diameter of the screw must be greater than 12 mm"* becomes an annoying manual and time wasteful task. In plus, other engineering specifications like manufacture specifications and their respective relationships evaluation, which somehow could be computationally aided, still rely on a human based operation (prevailing engineering experience basis to knowledge re-use).

Illustrating the second example, when purchasing items through electronic commerce, several product specifications are defined under semantic classes or in an imprecise way on purpose: to ask for a product's class, or for a nominal value property (like in the case of a mechanical component) not necessarily means that the product will have such final specification. In fact, manufacture specifications, whilst based on such broad definition, are far refined in order to guarantee product final commitments. But the search of the component is still made on basis of the imprecise nominal characteristics, semantically defined to human interpretation purposes.

Therefore, although STEP product data model remains the key standard for the recording and exchange of information about the composition, definition and shape of products, it is not entirely adequate for the suitable connection with knowledge based databases or publication of product information on the Web, which are using new technologies such as XML, RDF or OWL.

IV.2.1. Ontology concept

Ontologies have become a very popular form of knowledge storing and sharing within the last 20 years due to its broad application and many advantages over other forms of knowledge sharing. Recently, Artificial intelligence community adopted Ontologies as a knowledge modeling technique [50] to specify meanings in a formal way. The ability to create and operate on domain specific vocabulary and knowledge has been of interest not only to the Computer Science community, but also to industry and engineering community committed with interoperability's issues.

Asunción Gómez-Pérez [51] gives a complete insight and review of evolution and theoretical foundations of ontologies, from its philosophic point of view towards ontological engineering developments.

By definition, *Ontology* is a branch of metaphysics concerned with the nature and relations of being. In the knowledge-sharing community, an ontology is a description (like a formal specification of a program) of the concepts and relationships that can exist for an agent or a community of agents [52]. The primary benefit of ontology to engineers is the ability to classify, organize, and share knowledge related with *product models (*and thus product data models*)*. This same principium may be used when applied to knowledge of manufacturing problems. Integrating design knowledge, along with the knowledge of manufacturing structures, has the potential to provide substantial benefits in the product's lifecycle enhancement.

Thus, an ontology provides a formal method for identifying and classifying knowledge; a common vocabulary of an area is established while the meanings of the terms and the relationships between them are defined, with different levels of formalisms.

Soergel [53] made some considerations advertising about the difference between classifications systems and ontologies. Sometimes the notion of ontology is not clear since simple taxonomies may be considered full ontologies [54]. Examples are UNSPSC, e-cl@ass and RosettaNet, proposals for standards on the e-commerce domain, and the Yahoo! Directory, a taxonomy for searching the web, which are partially considered ontologies [55] because they provide a consensual conceptualization of a given domain. Nowadays, ontology community distinguishes ontologies that are mainly taxonomies from ontologies that model the domain in a deeper way and provide more restrictions on domain semantics, and thus a better conceptualization. Therefore, lightweight ontologies include concepts, concept taxonomies, relationships between concepts, and properties that describe concepts, while heavyweight ontologies add axioms and constraints. Axioms and constraints clarify the intended meaning of the terms gathered on the ontology.

Several languages were developed to specify formal meanings in ontologies. The most known are OIL, DAML+OIL and Web Ontology Language (OWL for short), the last of which is currently a World Wide Web Consortium (W3C) recommendation [56]. OWL is based in Description Logics (DLs) family of logical formalism for knowledge representation. In the origin of the decision to base those languages on DLs was the requisite that key inference problems (such as ontology consistency) should be decidable, and hence the language should enable reasoning services to be provided during ontology design and deployment.

Since ontology matters computer science engineers and mechanical engineers care about machine tools, helpfully a small example enlighten the process of modeling knowledge through an ontology.

The conceptualization of a certain domain recurs to a set of concepts, their definition and their inter-relationships. Typically, the conceptualization of an ontology utilizes five modeling primitives: classes, relations, functions, axioms and instances.

Classes or concepts mean any event from a set of objects considering from semantic facet. It may include function, behavior, strategy or description of a work to aggregate the class. Considering the type of fasteners, it includes rivets, studs, bolts, screws, among others. The properties of concepts are the static semantic feature and unrelated with context. For fasteners, properties include e.g. type of the head, type of the thread, nominal diameter, or mechanical properties like tensile strength or torque.

Relations of concepts are the dynamic semantic feature. They determine the inter-relationship of one knowledge individual with others in the same context. In a relation "subclass-of" example of the fastener, it may be included pin, stud or bolt under the class threaded. Other relations may include part-of, kind-of, instance-of or attribute-of, among others.

Axioms mean true declaration. For example, the statements «A bolt exists» or «Every bolt is a fastener» are examples. *Formal* conceptualization means that the concept could be translated into some form of logic, usually first order logic. Thus, the previous informal sentences could be written in first order logic axiom as «there exists an x such that x is a bolt» and similarly the later becomes «for all x, if x is a bolt than x is a fastener».

```
<owl:Class rdf:ID="bolt">
  <rdfs:subClassOf>
    <owl:Class rdf:about="#Threaded"/>
  </rdfs:subClassOf>
</owl:Class>
<owl:Class rdf:about="#Threaded">
  <rdfs:subClassOf rdf:resource="#Fasteners"/>
  <owl:disjointWith rdf:resource="#Non_Threaded"/>
</owl:Class>
<owl:DatatypeProperty rdf:ID="has_Standard">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#bolt"/>
</owl:DatatypeProperty>
```

Figure 17 – Ontology language codification of a fastener conceptualization (OWL extract file).

Figure 17 corresponds to an extract of an OWL file where a bolt is classified and properties semantically identified. A more complete example could be found in Annex IV . Several works on engineering ontologies are currently being pursued by diverse Design and Manufacture communities.

Among many other examples, a Georgia Tech team is exploring product knowledge interoperability and life-cycle management through ontology-based methods [57]. In their work, they discuss the beginnings of formalizing a process of creating a product view federation from component federates to enable the reuse of knowledge. Previous ontology works have recognized the advantages of using an ontology for interoperability, specifically a "port ontology," which "formalizes the conceptualization of ports such that engineers and computer-aided design applications can reason about component connections and interactions in system configuration". The University of Maryland has also proposed the use of ontologies as a way to address a need for a common knowledge base that will facilitate interoperability between software applications. NIST recognized the ability of ontologies to "make explicit the semantics for the concepts used, rather than just relying on the syntax used to encode these concepts", exploiting this attribute to create a well-defined knowledge base by giving unambiguous definitions of product and process capabilities [35].

In Europe, several international projects explored ontologies as an enabler of interoperability performance increase of Enterprise modeling and Product model.

The presented work was developed in the scope of the Intelligent Manufacturing Systems (IMS) SMART-fm programme (www.ims.org) and European ATHENA project (www.athena-ip.org), under real industrial environments. The results are considering to be adopted by its members as a basis for their business and manufacturing activities, whenever ontology deployment for the representation of product's catalogues and parts has to be developed.

The problem addressed is not specific from a particular kind of industry. The described issues can be found in similar situations, and ontologies deployment can be envisaged as a contribution to improve interoperability. In the ATHENA project, proposals to contribute to solve the same class of problems are in study to be applied in the automotive, aeronautics and telecommunication industries, considering as a basis the framework of the present work.

IV.2.2. Methodologies for ontology development

Academic and industrial communities have been developing methodologies to guide in the process of ontology building. Based on Gómez-Pérez *et al* [51] [52], the author clustered the following set of characteristics to clarify ontology building methodologies, which are:

1) starting from scratch (S);

2) reengineering (Re), i.e. ontology building based on existing ones;

3) cooperative building (Co), i.e. the actors should be able to participate in the process;

4) merge methods (Me), i.e. ontology merging, integration, use and mapping.

Ontology merging aims to make a more general ontology about a subject, by gathering knowledge from several other ontologies in that same subject. *Ontology integration* reuses other ontologies, while, each integrated ontology is about a different domain other from the resulting ontology. *Ontology use* builds an application using one or more ontologies - there is no resulting ontology [58]. *Ontology mapping* is an activity that attempts to relate the vocabulary of two ontologies that share the same domain of discourse [59].

In Table I a set of available ontology building methodologies is presented. They are presented categorized by identified ontology building characteristics.

TABLE I – Ontology Building Methodologies Analysis

Methodologies					Tools	Languages
	S	Re	Co	Me		
Cyc	✓				KAON	OCML
Uschold & King	✓				OILed	**OWL**
Grüninger & Fox	✓				OntoEdit	**DAML+OIL**
KACTUS	✓				WebODE	OIL
METHONTOLOGY	✓	✓			**Protégé**	**RDF(S)**
SENSUS	✓				Ontolingua	**XML**
CO4			✓		Server	KIF
(KA)^2			✓		Ontosaurus	Ontolingua
On-To-Knowledge	✓				WebOnto	**OKBC**
FCA-merge				✓	VOID	LOOM
PROMPT				✓	Apollo	**FLogic**
ONIONS				✓	CODE4	
UPON	✓					
MENTOR	✓	✓	✓	✓		

Legend: ——▶ method. direct relation; - - -▶ method. indirect relation;
—— languages used by the tool
S – Scratch; Re – Reengineering; Co – Cooperative; Me – Merge

The *CYC* methodology consists on extracting, by hand, common sense knowledge that is implicit in different sources. Once enough knowledge in the ontology is available, new common sense knowledge can be acquired either using natural language or machine learning tools [60].

Uschold & King proposed some general steps to develop ontologies, which are: to identify the purpose; to capture the concepts and the relationships among these concepts and the terms used to denote both of them; to evaluate and to document the ontology [61].

Grüninger & Fox methodology is based on identifying the main scenarios and the competency questions, followed by extracting relevant concepts and relations and formalizing them [62].

In the *KACTUS* project an ontology methodology was proposed, which building process is based on a Knowledge Base (KB) application, by means of an abstraction process [63].

Methontology is a methodology for building ontologies either from scratch, or by a re-engineering process. The *Methontology* framework enables the construction of ontologies at the knowledge level. It includes: identification of the main ontology development process activities (i.e., evaluation, configuration management, conceptualization, integration, implementation, etc.) [54].

Sensus is a methodology that follows a top-down approach for deriving domain specific ontologies from huge ones. The process is based on a first manual identification of terms that are relevant to a particular domain and a consequently automatic extraction of the related hierarchical structure of the previous identified terms. The result is a skeletal foundation for a KB [64].

The *CO4* methodology foundation is based on a protocol defined from identified problems concerning collaborative construction of ontologies. The goal is to reach consensus among several KBs and it is based on the main idea that people can discuss and commit about the knowledge introduced in the KBs. These KBs are built to be shared, and they have consensual knowledge, hence they can be considered ontologies [65].

The *(KA)2* methodology is based on ontologies development in a joint effort by a group of people at different locations using the same templates and language. The process is based on ontology coordinating agents which distribute a template among the ontology topic agents (experts in different topics). Once the ontology coordinating agents got all the portions of the ontologies from the ontology topic agents, they integrate them, activity that benefits from the presence of a common pattern [66].

The *On-To-Knowledge* methodology consists on generating editable ontologies automatically from natural language documents followed by human interaction in order to ensure the quality of the results [54].

The method of *FCA-Merge* is guided by application-specific instances of the given source ontologies that are to be merged. Natural language processing and formal concept analysis techniques are applied, in order to derive a lattice of concepts. The generated result is then explored and transformed into the merged ontology with human interaction [67].

PROMPT is an algorithm that provides a semi-automatic approach to ontology merging and alignment. PROMPT performs some tasks automatically and guides the user in performing other tasks for which his intervention is required [68].

ONIONS is a methodology for conceptual analysis and ontological integration or merging of terminologies. *ONIONS* aims to provide extensive axiomatization, clear semantics, and ontological depth in the domain terminologies that are to be integrated or merged [69].

UPON is a Unified Process for Ontology Building based on workflows of iterations related to the ontology building requirements, analysis, design, implementation and testing [70].

In general, there is no correspondence between ontology building methodologies and tools, although there is a few that has a direct tool which implements its related methodology process as it is shown in the Table I. These have been identified as the most known, but it is quite provable that many others are under development by active computer science community.

IV.2.3. Ontologies proliferation and integration issues

Competition keeps increasing and organizations are considering improving their position in the market performing strategic partnerships [71; 72]. The formation of cooperation and collaboration alliances between several small organizations is proving, in multiple cases, to be more efficient and competitive by comparison with big companies. Although explained in Chapter III, this is typically what leads companies to join efforts to survive in very evolutionary and dynamic markets [73].

However, partnerships cause some problems mainly in integrating Product Life Cycle phases, since manufacturers, distributors, designers, retailers, warehouses, often have their proprietary solutions which are, typically, not interoperable one with another [74]. As in manufacturing, the necessary incorporation of different brought-in parts requires detailed data check and update by different teams, using heterogeneous applications to keep valid the initial design conditions and maintain the product assembly consistency. This task is often facilitated with computational tools, and the components' data representation should be of common understanding, to reduce re-work and time consuming during catalogue examination.

Indeed, due to the worldwide number of existing catalogue components and diversity of teams' culture, only in very specific situations one supplier would adopt the same terminology of a manufacturer or vice versa. That was supposed to be solved through the standardization, both of the component and the main variables designation, though only the first succeed. To impose a unique terminology and classification would be a solution. However, those suppliers not adhering to it most probably would be ignored, which is not a favorable business situation, not permitting for instance an open selection of the supplier.

Moreover, in the advent of internet and e-business electronic data share, the translation of previous paper representations in e-catalogues did not contribute to solve the mismatches and close the gap.

Several initiatives were taken to address this issue, like ISO10303 STEP technology in components computational representation. STEP product data model, e.g. through AP214, enable the exchange of well-defined geometric and topological features and some additional product information. Although a powerful language to model engineering data (EXPRESS) is used in STEP data models, it is not suitable to bring semantic skills into the model, thus knowledge is not properly captured in STEP data structures.

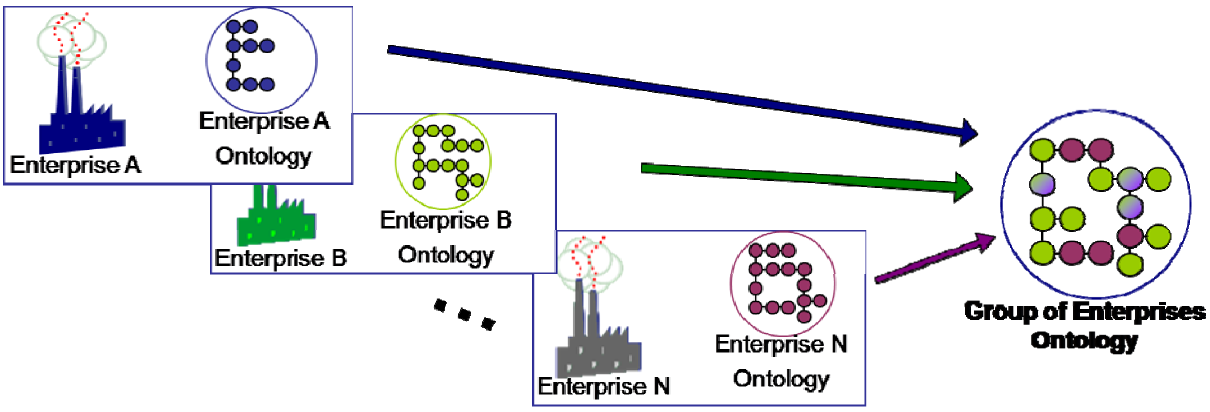


Figure 18 – Reference ontology for a group of enterprises.

More recently, the development of ontologies, as promising techniques with capabilities to solve semantic issues, has been addressed by important companies and SMEs. Thus, each company is struggling to develop competencies at this ontological level, but inevitably different perspectives will lead to different final results, and achieving different ontology in the same business domain is the reality.

For that reason, to envisage forcing stakeholders to adopt a same ontology, even if it is based on standards, does not work in most of the cases. Thus, an ideal solution would be to keep the terminology and classification in use by each one, and to use a reference ontology to be the mediator in the communications between them (Figure 18).

Additionally, the introduction of a new reference ontology would enrich the community and each enterprise should feel motivated to be part of the group, with the possibility to keep their own definitions and being offered the enhancement of their own ontologies. This concept will be addressed in the separate sub-chapter *V.3 - The MENTOR methodology.*

IV.2.4. Ontology support in Product Life Cycle

After the scenario envisaged in Chapter II, precise data exchange between internal and external production agents plays a key role, where an initial product data model must be instantiated during the early design and tooling phases, being updated along the PLC.

To have this data flow accurate, data models and processes need to be interoperable. However, this situation has been identified as difficult to achieve, because typically there are many different software applications in use, each one adopting its own data structure and semantics [9] [29] [75] [76].

Additionally, when developing a product each participant team normally has its specific method of work and self-containing language. This fact does not result in a flawless interaction with others. For example, it is frequent to find the situation where in different PLC stages the use of dissimilar vocabulary addresses exactly the same component [77].

It results that, in general, the information managed in each PLC stage regards to different natures, classifications, levels of access and detail.

For example, the design team may entitle a bolt using one expression that, to prevent design errors, needs accurate information regarding the component. However, such detailed information may be valueless to the maintenance team, which for a successful part replacement, only needs the reference or the component genre identification. This means that design team “views” the same information in a distinct manner and with a different level of detail than the maintenance team.

Another example is when in front of the exactly same product model certain features are differently perceived according to the team background or discipline dealing with it. For example, Figure 19 shows the same feature, though highlighting the interpretation that respectively Design and Manufacturing team may have. Even the nomenclature when referring to the same shape differs: a rib to a Design team environment corresponds to a two steps milling operation of the Manufacture team.

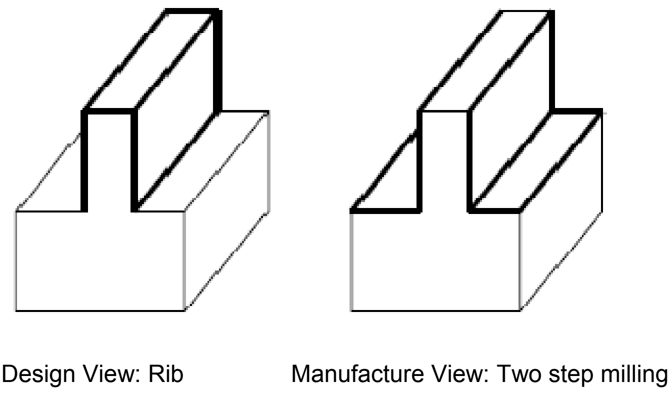


Figure 19 – Interpretation of the same feature according to the different team's background.

Figure 20 depicts a generalization of this circumstance along some principal PLC stages, where different "views" are identified on top of a same taxonomy, highlighting different viewpoints on the same class of information, e.g., with semantic and vocabulary variants.

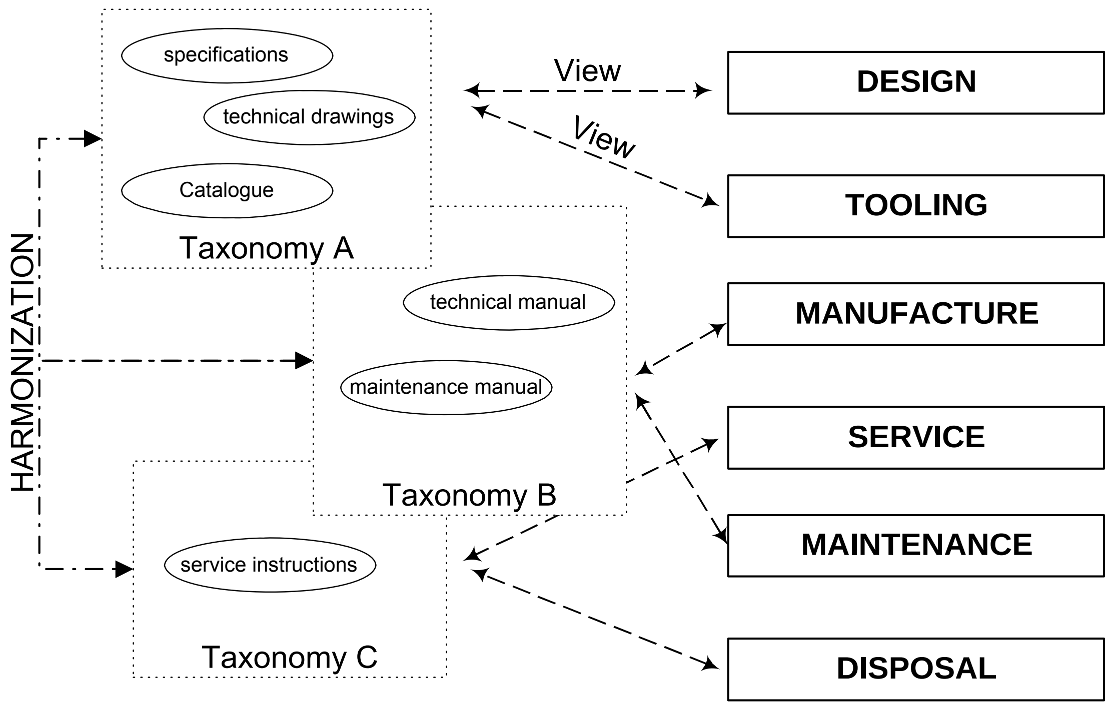


Figure 20 – "Views" according to the different taxonomies in use by PLC stages.

Although not represented in this figure, additional "views" show up when considering external agents, e.g., suppliers, when bringing their own taxonomies into the system. Thus, the scenario of different taxonomies identified within and across the PLC stages, internally and externally to the organizations, makes this problem complex.

This scenario of broad heterogeneity motivated the proposal of a framework to enhance Enterprise and Product modeling integration. An illustrative industrial example is described, addressing the manufacturers when exploring different suppliers of mechanical components. The complexity of the problem is noticed when singular difficulties arise in the moment that clients and suppliers with different qualified profile and background attempt to use and exchange catalogue components data.

IV.3. Ontological harmonization within *PLM systems*

Ontology is the study of the categories of things within a domain and reflects a view of a segment of the reality. As early defined in this Chapter, definition comes from philosophy and provides a logical framework for research on knowledge representation, embracing definition, classification and relationships of concepts [78].

In this context, two or more communities (e.g., organizations, teams), operating in the same domain may use different terminologies and have different views on the same concept, leading to different underlying ontologies, and consequently conducting to problems of interoperability. At a first level this problem comes out in the communication between humans, then between humans and computer systems, and finally between computer systems [61] [62].

IV.3.1. PLM heterogeneity

For example, when a client talks with suppliers searching for a specific component, they need all to understand each other. If for any reason this is not the case, humans are able to reasoning and combine their knowledge attempting to converge to a common understanding, and hence communicate.

In opposition to this interactive and intelligent human to human process, computer systems communicate under a well established syntax, through rigid communication protocols. However, the inclusion of semantics in the communication protocol under a well established classification mechanism, making use of knowledge modeling components described according established semantic representation paradigms, complements the information exchanged contributing for an enhanced understanding between the systems [51].

PLM Systems, as the core computational support to PLC activities, are required to manage this environment of broad heterogeneity, and, in a certain way, to replace the human interaction process mentioned before.

Therefore, an interoperable system that seamlessly communicates and understands each other requires the comprehensive understanding of the meaning of the data exchanged within the domains involved. This can be realized, if the communication process is supported by an ontology developed under global consensus [62].

To assist solving this sort of problems, Radeke [79], Davulcu [80] and Erwan Berton [81] point towards the use of a single ontology framework. However, in the presented situation of broad heterogeneity, where many taxonomies and semantics can be found within the product life cycle, to achieve seamless interoperability within and between its stages will require a common agreement of the complete system's information structure and semantics. Only in this way it will be possible to unambiguously describe the meaning of the whole data instantiated in the enterprise models, and exchange it between the applications along the PLC stages.

Kishore, Zhang and Ramesh [82] divided the scope of the ontologies into three abstraction levels, i.e. 1) the universe of discourse, 2) a domain of interest, and 3) an instance of the domain of interest. These three levels are focused respectively on the 1) general purpose ontologies, on the 2) underlying frameworks and on 3) particular company's systems.

Also, Noy [83] [84], McGuinness [85], Mitra [86] and Klein [87] propose a set of methodologies and software tools to support the development and harmonization of ontologies, essentially when its use becomes more prevalent. Three perspectives have been identified to manage such harmonization by means of

1) merging [88] [89],
2) aligning [68] and
3) integrating [88] procedures.

The work presented in the following Chapter V addresses applied research in the underlying frameworks and their respective instantiation on particular company's systems. The combination of the many ontological worlds towards a harmonized one to get an enhanced stage of global common understanding is the envisaged best solution.

This harmonized ontology will belong to the teams of involved participants, reflecting a unique taxonomy of concepts complemented with detailed common descriptions and fundamental knowledge. It will represent the unambiguous description of the information semantics to be exchanged between the participants, achieved through combination of each taxonomy structure and concepts that were developed independently.

The use of different taxonomies by the various agents when referencing to a same component is identified as a source of systematic errors and misunderstandings. The methodology proposed in this work assists organizations to solve this problem in nowadays traditional industrial working procedures, anticipating the use of web tools to support the implementation, whilst preserving manufacturer's departments' culture and suppliers' independence.

Having this problem solved, manufacturers and suppliers can save costs (e.g., buying cheaper a same component), additional work (e.g., easier procedure to identify a component) and time schedule delays (e.g., faster to get a solution), and set their business open to new opportunities (e.g., catalogues are universally understood).

IV.3.2. Converging by means of harmonization

To obtain a consensual model, it is necessary to classify and merge the concepts from the different sources within the domain of applicability, describing them in a unique harmonized structure of classes, attributes, relationships, knowledge components and definitions [51]. An example of a traditional mechanical component (*a hexagonal bolt*) and the main concepts and definition associated will help to clarify some harmonization aspects.

As exemplified in Figure 21, the classification of *Bolt*'s properties of *Supplier A* differs from a *Supplier n*. Also, the attributes' definition are different for the same *Bolt* property addressed, i.e., both bolt head properties *flat_width* or *s*, respectively in taxonomy of *Supplier A* and *Supplier n*, specify exactly the same bolt head dimension. Other similar occurrences can be identified within this same example.

Through a combining procedure, the harmonized classification is defined, structuring the various suppliers' information from different sources and for diverse product categories [90].

Figure 22 describes the three phases of the method to achieve a harmonized ontology for catalogue representation. For exemplification, it is used a sub-set of the catalogue (right hand-side of the figure), with focus on the component *Bolt*.

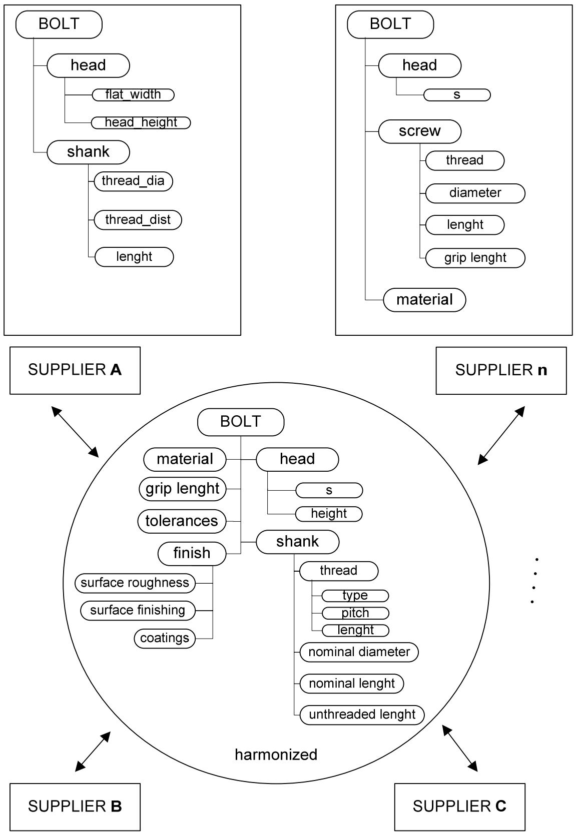


Figure 21 – Example of a scenario for harmonized classification.

The first phase, identified as *ad-hoc catalogue representation*, refers to the representation of catalogues as they are usually found in the market. Here, the definition of catalogue has casual categorization or nomenclature, belonging to an incipient phase of a taxonomy development which terminology typically does not follow the standards.

In the *structured catalogue representation* phase, catalogue's components are classified and organized, conferring logical structure and semantic contents to the catalogue representation. The adoption of standard-based specifications, versus proprietary, should be encouraged to be taken on this phase.

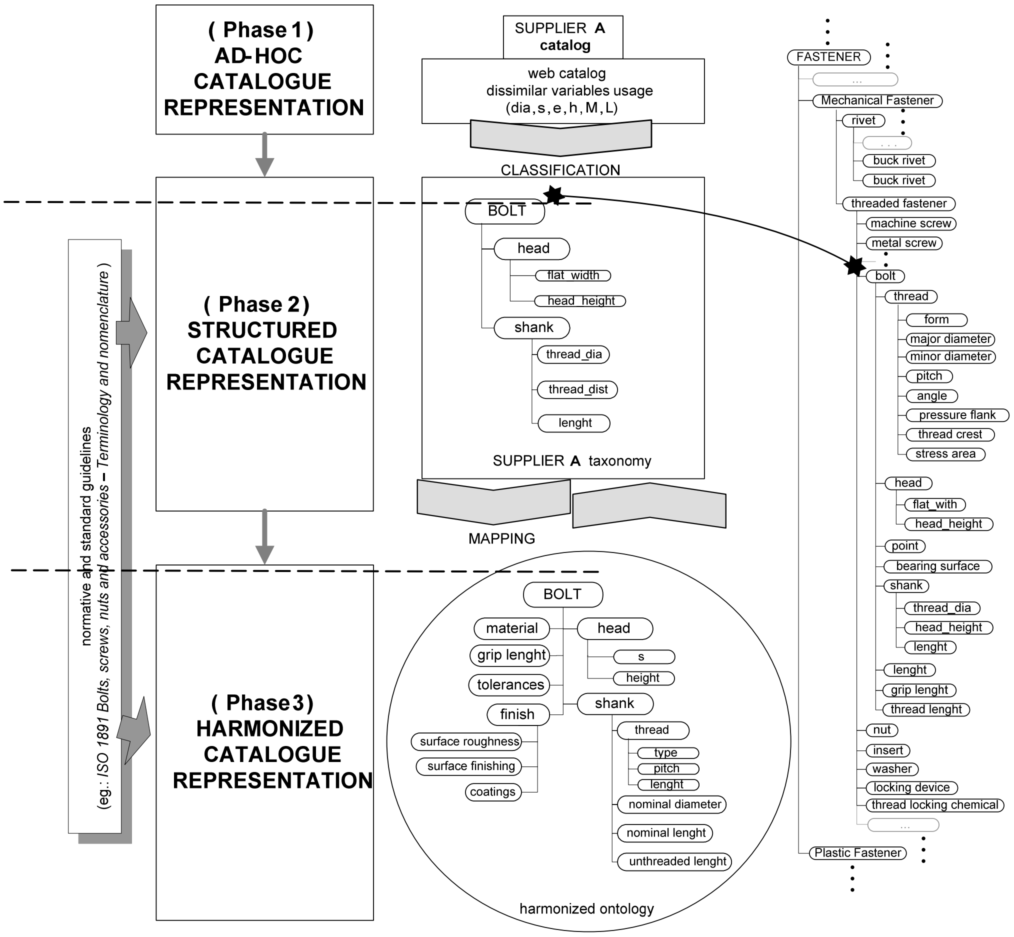


Figure 22 – Phases to achieve a harmonized catalogue representation - focus on "*Bolt*".

In the case of catalogues of *Bolts* it is recommended to be used as a reference the standard previously defined and developed by ISO (e.g., *ISO 225:1983 Fasteners – Bolts, screws, studs and nuts –Symbols and designations of dimensions*). However, although this facilitates the harmonization of the catalogues, companies typically react to change its own culture and methods of work, and refrain to adopt the standard in favor of developing a proprietary structured catalogue. Nevertheless, to achieve a structured representation is a major advance in the representation of their catalogues, offering novel possibilities for search and analysis of its components.

During the third and last phase, identified as *harmonized catalogue representation*, it is developed an agreement among other agents in the domain of application. Thus, catalogues' representation are harmonized and complemented with semantics, towards a consensual common ontology. The methodology proposed to develop this harmonization is described with further detail in next Chapter and next section outlooks harmonization methodology.

The lost of supplier's organizational identity could be a concern when developing the *harmonized catalogue representation*. However, this is not the case, thought each catalogue's owner should keep bidirectional mapping rules between its own structure and the harmonized one, enabling organizations to keep using its own practices. A Mediator may be considered as explained in next Chapter.

IV.3.3. Methodology for harmonization of ontologies

This sub-section briefly explains the principles for a methodology to develop harmonized ontologies, as depicted in Figure 23. The focus of this methodology was on catalogues of mechanical components (specifically the Procurement valuable PLC phase), but methodology architecture took on consideration his future deployment to other PLC Phases.

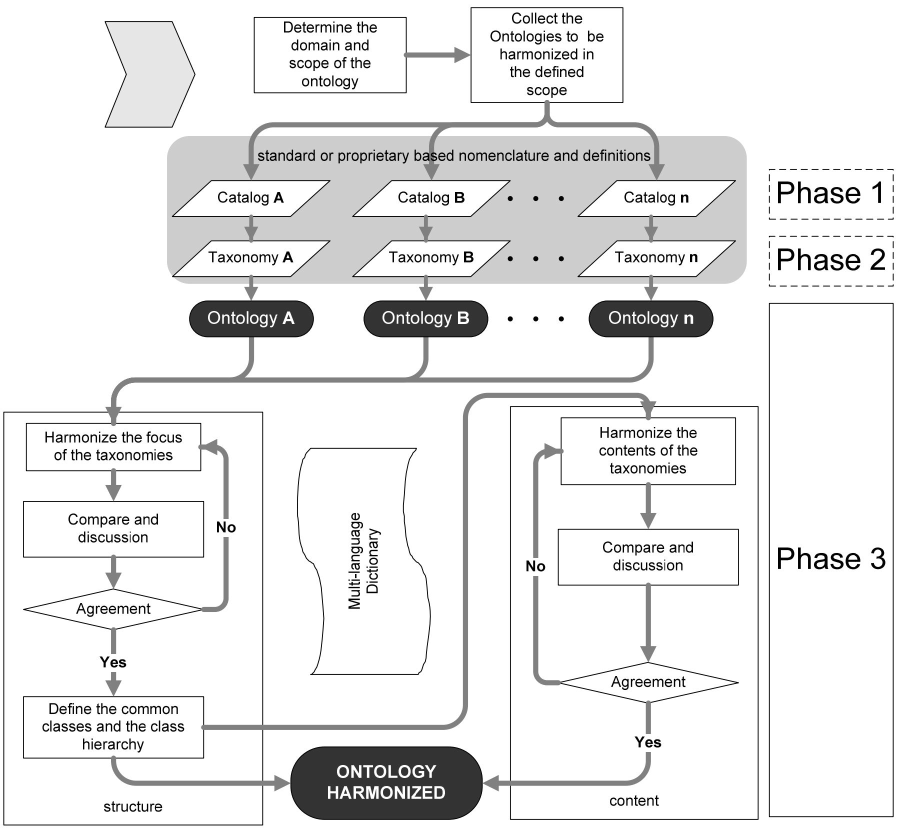


Figure 23 – Methodology for harmonization of ontologies.

This methodology supports the progress from the phase 2 to the phase 3 (Figure 22 above) to complete the harmonization for representation of catalogues.

The entry point for the methodology is the identification of the domain and scope of the ontologies to harmonize. Each supplier takes the catalogues in such range, and develops the respective taxonomy and inherent semantics according to the procedures explained in previous section (Phases 1 and 2).

The choice of the taxonomy's scope and the subdivision of subjects rely on the market itself, with influence of the pre-existing catalogues and standardization initiatives. The development of each own supplier's ontology, can be done based on the method proposed by Noy [68].

Having the set of ontologies available (Phase 2), the harmonization process can start.

Ontology reconciliation is a human-mediated process, supported by specialized software tools, addressing the merging, aligning and integrating procedures for ontology harmonization. The management of ontology reconciliation and the resolution of most of the ontological mismatches require direct human involvement to identify, for instance, unique concepts and concepts that are similar in meaning but have different names or structures.

Indeed, mismatches between individual ontologies can take place at conceptual, terminological, taxonomical, definitional, or even, at syntactic level. Thus, correspondences, i.e., mapping, between concepts and knowledge have to be established, identified gaps bridged, and acknowledged overlaps matched.

Phase 3 is divided in two sequential procedures, to be mediated by humans and assisted by software tools for ontology reconciliation, e.g., Prompt [68], Chimaera [85], ONION [86], OntoView [87], GLUE [91], OBSERVER [92].

Initially, this phase works towards an agreement on the ontology's common structure, i.e., hierarchy and relationship between classes (left-hand side of Figure 23), and later on its contents, i.e., knowledge modeling components for semantic analysis and reasoning support (right hand side of Figure 23). Along Phase 3, the reconciliation tools are used to support the human decisions towards the harmonized ontology. Because it is unlikely that one single tool adequately handles all aspects of ontology harmonization, the Hameed's workbench should be evaluated for the appropriate tool selection in each stage of the reconciliation process [93].

With the boundaries of each taxonomy on hand identified, they are discussed in workshops until the harmonizing team reaches a common agreement on the focus for the foreseen combined taxonomy. The definition of the harmonized taxonomy's scope and the subdivision of subjects should be leaded by the users, and guided by the market itself with influence of the pre-existing catalogues, standards and expertise of the involved agents. Concentrated on the agreed focus, the common classes and respective classification are defined, and the structure of the harmonized ontology established.

Following a similar procedure, and guided by the established harmonized structure, the knowledge modeling components in place are compared and discussed until agreement how they might be combined and incorporated. When accomplished, the harmonized ontology is thus established together with the mapping tables describing the ontological relationships between the harmonized and each one of the individual ontologies.

Semantic difficulties related to the natural language of the potential users of the harmonized ontology likely happen. To assist on it, the ontology is complemented with a multi-language dictionary where a set of normalized tokens gives the reference to the corresponding concepts and definitions in different native languages [94].

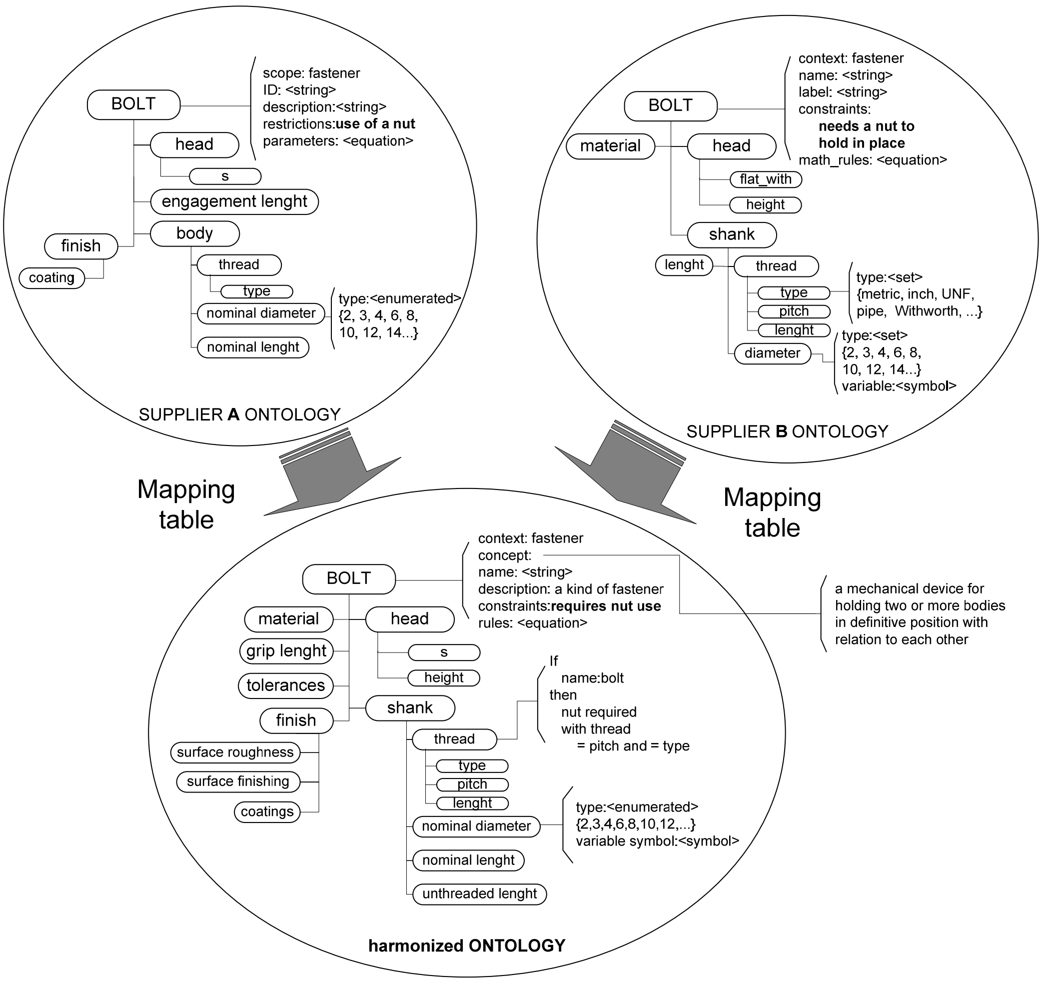


Figure 24 – Example of harmonization between suppliers.

Figure 24 exemplifies one harmonization of individual ontologies between two suppliers, with focus on *Bolt*. In this case, the terminology, concepts and knowledge modeling components of each individual ontology were harmonized through inter-partners consensual procedure, according to the proposed methodology.

In this example, it is illustrated that *Supplier A's nominal diameter (enumerated: 2,3,4,…)* is harmonized with the *Supplier B's diameter (set: 2,3,4,…)*, resulting in the *nominal diameter (enumerated:2,3,4,…)* representation. Also, the *Supplier A's restriction (use of a nut)* is harmonized with the *Supplier B's constraint (needs a nut to hold in place)*, resulting in the *constraint (requires nut use)*. As well, a new rule was added to the harmonized ontology, stating that a *bolt always requires a nut with equal thread's pitch and thread's type*.

The dimension of the resultant ontology, as a harmonization of different ontologies, will not necessary reflect the arithmetic sum of the defined concepts in each of the individual inputs. Although most probably it will be larger than the most complete individual one, thought it could be expanded with new concepts and content richness, very often it is essentially the same considering it is defined to operate in the same scope and only direct mapping between concepts is needed to be established.

As well, each supplier's ontology will not get expanded after harmonization, having in the mapping table the characterization of the ontological relationships with the harmonized one and a Mediator as coordinator of such changes.

The harmonized ontology, when established and accepted by a substantial critical mass of users, shall result in a standard. Industrial associations, business consortia and users' groups shall assume the leadership to maintain and keep it updated, and work close to the standardization bodies to develop the necessary normalization activities. The funStep IG (hhtp://www.funstep.org) is the example of a user's group that has been working for the furniture industry in this way.

IV.4. Outlook

The information managed by one organization habitually embraces situations of broad heterogeneity, where concepts need to be handled under different structures, knowledge modeling components and levels of access and detail.

For an enterprise to achieve seamless information interoperability, it requires a common agreement of the global information system's structure and semantics. Only in this way it will be possible to unambiguously describe, internally and externally to the organization, the complete meaning of the data instantiated and exchanged through the enterprise models.

To contribute to get such stage of global common understanding, this chapter proposes a methodology that combines the many ontological worlds in place, i.e., the instances of domain, developing a harmonized ontology aiming to represent a domain of discourse. To obtain this consensual model, it is necessary to classify and combine the concepts from the different sources within the domain of applicability, describing them in a unique harmonized hierarchal structure of classes and definitions.

To simulate the proposed methodology, industrial scenarios have been experimented addressing the interoperability problem between manufacturers and suppliers of mechanical components, when they face communication difficulties generally resulting from the lack of knowledge regarding data representation and semantics of the other interlocutor.

When the methodology was applied to combine a set of selected catalogues of components, it resulted in a harmonized ontology described along with the definition of the relationships between the manufacturer's component classification and those defined by each supplier's.

It is recommended that the harmonized ontology could be implemented in an open-source software platform. The platform used in current work is generic and open, providing web-based functionalities. With it, the mediator got a computer based interface between the information from the users (e.g., manufacturer's design team) and suppliers' catalogues.

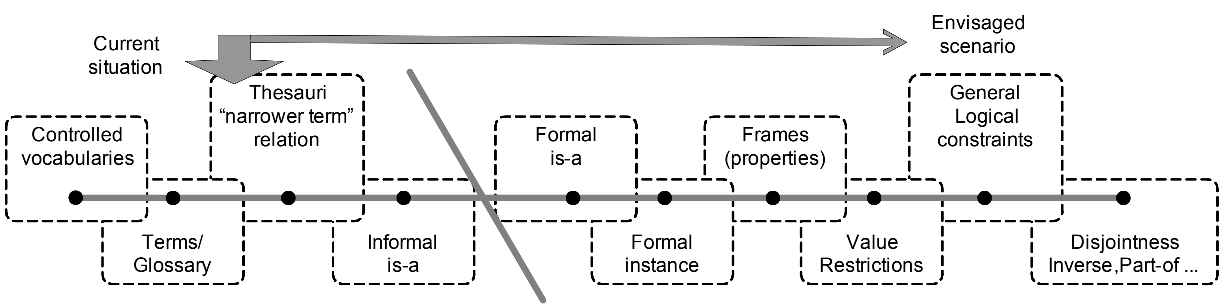


Figure 25 – Impact of the work's contribution, according to the Lassila and McGuinness categorization.

The platform was experimented with encouraging results under the scenarios described, and the proposed framework was recognized innovative and of economical relevance for the participating users, as an enabler for new opportunities to identify and select components, and for the implementation of faster and more accurate management procedures.

Lassila and McGuinness [55] classify ontologies according to the information the ontology needs to express the richness of its internal structure. Considering the current SMEs industrial scenarios, the typical current situation for representation of industrial catalogues of components falls in the Terms/Glossary category.

With the contribution of the work, the envisaged scenario is the one where the various PLC information sources (like catalogues, CAD files) will be assisted by ontologies that may place restrictions, expressing general logical constraints. Figure 25 depicts the identified impact of the work's contribution, according to the Lassila and McGuinness categorization.

Chapter V

V. Framework for Ontology harmonization

The availability of a computational product representation where geometric and technical product data may be integrated with semantic properties makes easier the integration of services and applications among multiple production related computational tools. This is of great importance in the advent of dissemination of webservices world wide through internet, assisting product life cycle stages. Moreover the co-existence of different backgrounds, expertise and knowledge skills in the teams involved in Design and Manufacturing have been explored due to their added value in certain cases.

Therefore, ontologies facilitate the use and exchange of data, information and knowledge among people and organizations, towards intelligent systems interoperability. Nevertheless, concurrent initiatives on distributed and heterogeneous systems originated more than one ontology development, and in consequence various parties with different ontologies often do not understand each other. This work proposes a methodology to support the development of a common reference ontology within manufacturing context for a group of enterprises sharing this domain.

This methodology is based on the concept of Mediator Ontology, which assists the semantic transformations between each enterprise's ontology and the referential one. This methodology enables each organization to keep its own terminology, glossary and ontological structures, providing seamless communication and interaction with the others.

V.1. Manufacture: the thread archetype

In mechanical environments, the troubles caused by physical components' lack of interoperability are well known. One of the classical physical compatibility problems regards to the thread interconnection. A solution for this problem resulted in the development of an international standard addressing this physical level of integration. Nowadays, the high interchangeable level of metric screws and nuts turns this issue almost unperceived.

Considering a standard thread, when a M4x10 or M12x25 is stated in mechanical terminology, a metric thread shape is meant and nominal variables are exactly defined. In fact, the profile and dimensions of metric threads are defined by the international standard ISO 68 Part 1:1998 General purpose screw threads [95], as pictured in Figure 26.

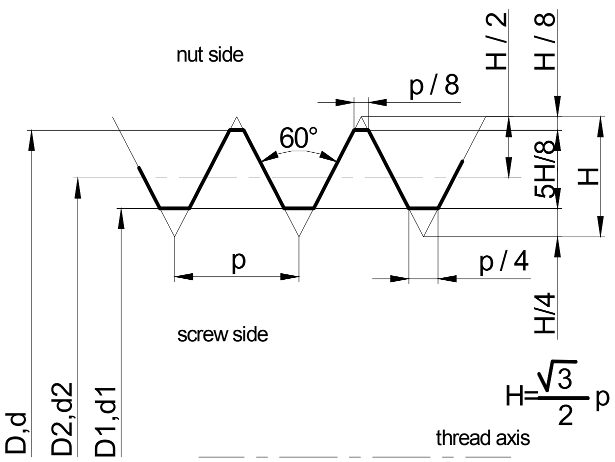


Figure 26 – Standard ISO variables defining a metric threaded surface.

No matter what kind of industry being addressed, mechanical, chemical, furniture, polymer or any other industrial field, an ISO metric thread designation will have a unique matching physical metric thread. According to the PLC model presented in the Chapter II, computational integration for the thread data should be driven by ISO standard parameters and made available to the entire range of stages, from early design to subsequent life cycle phases.

Nevertheless, engineering staff in companies (from manufacturers to retailers), did not adopt the standard terminology in fully extension due to multiple reasons. Slight adaptations to proprietary expressions in practice in each job-shop floor and complementary information or nomenclature with more meaning inside each corporation as well are only two among many other explanations in the origin of different terminology.

The development of a standard-based thread data model should enhance thread shape knowledge availability to the multiple PLC phases, from design and manufacture to maintenance or disposal. Whatever information level being addressed, from simple weight to parametrical data, seamlessly incorporation into wide-ranging product model will be possible, thus bringing different operational levels to PLC activities.

Product shape is a result of all PLC activities, and the best way to get some efficiency is by re-using previous data and knowledge, sometimes imperative in the first product shape iteration.

Knowledge capture has been the target point of recent design research trends. The development of various modular product models, regarding integration capabilities, will split the overall problem facilitating partial solutions, although requiring advanced skills for afterwards integration. Thus, the development of a modular thread model may enable segmented information and knowledge capture mechanisms, dynamically focused to be worked at different PLC levels.

Nevertheless, the thread data model development by different companies or organizations may lead to the existence of dissimilar data models, with unlike variables, relationships or properties definition. Therefore, difficulties could be expectable when sharing or exchanging data where the same original concept will be materialized or implemented with unequal data structures and entities.

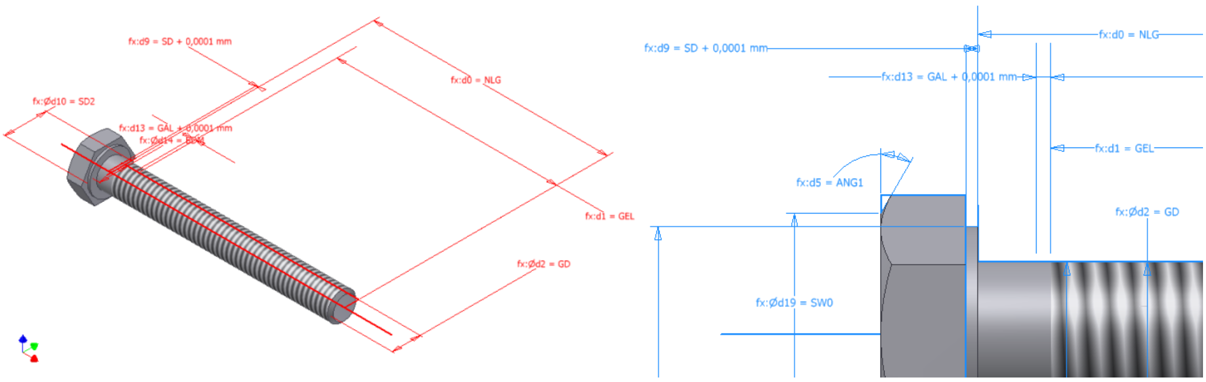


Figure 27 – Variables used by computational tool to model standardized ISO 4017 bolt (see Table II).

TABLE II – Variables of an ISO 4017 Bolt stored in a Library's product model.

Variable	Label	Expression
NND	Nominal Diameter	
NLG	Nominal Length	
SW	Width Across Flats	
KOH	Head Height	
SDHG	Shank Length without Threads	
RR	Radius	
GEL	Thread Length	
E0	Width Across Corner	
SD2	Shoulder Diameter	
SD	Radius	
BDM	Shank Diameter min.	{NND}
GD	Thread Diameter	{NND}
SPD	Cone Point Diameter	{NND}-(1.2269*{PTC})
SDH	Shoulder Diameter	{SD2}
SW0	Width Across Flats	{SW}
SIZE	Designation	M & {NND} & "x" & {NLG}
ANG1	Chamfer angle	30
THREADDIA	Nominal Diameter	
TPU	Thread per Unit	1 mm/{PTC}
THREADESC	Thread description	M & {THREADDIA} & "x" & {PTC}
GD1	Pitch Diameter	{THREADDIA}-(1.2269*{PTC})
GUL	Thread Run-out 45	1.2269*{PTC}/2
GAL	Thread Run-out 45	1.2269*{PTC}/2
PTC	Pitch	
TOA	Thread Open Angle	30
THREADCLASSEXT	Class	6g
THREADTYPE	Thread Type	ISO Metric profile
DESIGNATION	Size Designation	M & {NND} & " x " & {NLG}
FILENAME	File Name	ISO 4017 & " - " & "M" & {NND} & " x " & {NLG} & "ISO"
MATERIAL	Material	Stainless Steel, 440C
PARTNUMBER	Part Number	ISO 4017 & " - " & "M" & {NND} & " x " & {NLG}
SIZE_SEL	Thread description	M & {NND}
KLG	Grip Length	{NLG}-{GEL}

Just to describe an example, Table II lists the variables used by a Solid Modeler as stored in the Components Library database of an ISO 4017 metric bolt (see Figure 27). Comparing standardized variables definition depicted on Figure 26 with the corresponding ones on Table II, it becomes evident that not a single one matches the standard. Any further investigation surely may unveil the many reasons beyond this fact, although without contribution to mismatch's solution.

The adoption of ISO 10303 [32] may discipline the thread data model development, bringing the needed data integration. Also, the reuse of standard model schemata, like integrated resources and application protocols (APs), in a multilevel modular architecture should improve consistency and compatibility, through modular application protocols.

However, without a systematic method to dynamically integrate the specific modular models with each other along the PLC stages (e.g., the thread shape), an universal interoperable Product Life Cycle scenario is not envisaged to be achieved [96]. For instance, the exchange of a STEP file created after the same component of Figure 27, using AP214 (see Annex I), truncates those variables by addressing mainly geometric data. Those variables' inclusion in the Part 21 file of the component may be clearly considered out of scope of AP 214, however recent developments on others APs like parametrics do not accommodate a solution with entire PLC perspective.

V.2. E-Procurement: a classic mechanical bolt survey

Along the PLC phases, several stages can be identified where the search for products in catalogues is necessary. A typical situation is whenever it is needed the replacement of a component during manufacturing or maintenance. As well, when designers intend to provide alternative solutions, preventing single supplier dependency and reducing the manufacturing costs, the inclusion of alternative parts in the design of a product is a wise approach of management requiring the handling of a range of catalogues.

The necessary incorporation of different brought-in parts requires detailed data check and update by different teams, to keep valid the initial design conditions and maintain the product assembly consistency. This task is often facilitated with computational tools, and the components' data representation should be of common understanding, to reduce re--work and time consuming during catalogue examination.

Considering the specific example of searching for a classic mechanical bolt, many catalogues in paper or in digital format can be found. Nowadays, with the Internet browsing potentialities, a popular engine like *Google* can immediately find a large number of links for online catalogues of hexagonal head type bolts.

Amazingly, although all these catalogues reference to the same type of physical component, each one usually represents their specifications in different formats and with heterogeneous contents and classification. In fact, different catalogues tend to adopt different variables, unlike coding fields (though equivalent) or widespread designations, most of them diverging from available advisory ISO standard designations.

In practice, the International Standards *ISO 1891:1979 Bolts, screws, nuts and accessories – Terminology and nomenclature, ISO 225:1983 Fasteners – Bolts, screws, studs and nuts – Symbols and designations of dimensions* and *ISO 4017:1999 Hexagon head screws* specify terms, dimensions, tolerances and material requirements, including metric coarse threads and diameters for hexagon head screws. These standards represent a first worldwide attempt to uniform variables, structure and designation of fastener's components and hexagon head screws. However, they describe them in a glossary style, complemented with some illustrative figures, without any ontological principles underneath. An extract of ISO 225 standard is depicted in Figure 28.

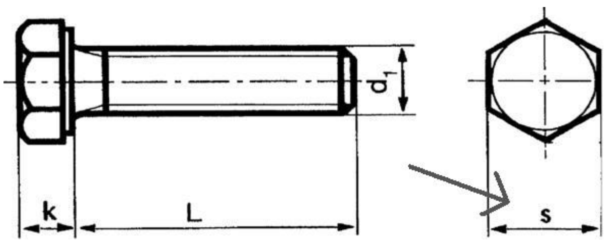


Figure 28 – Main bolt variables definition according ISO 225 [95].

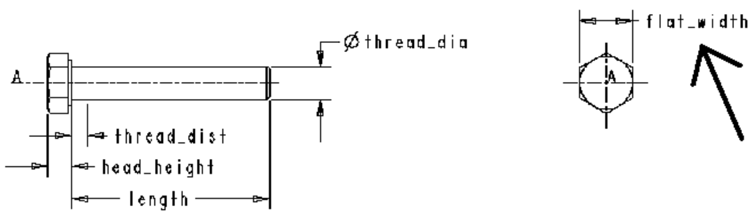
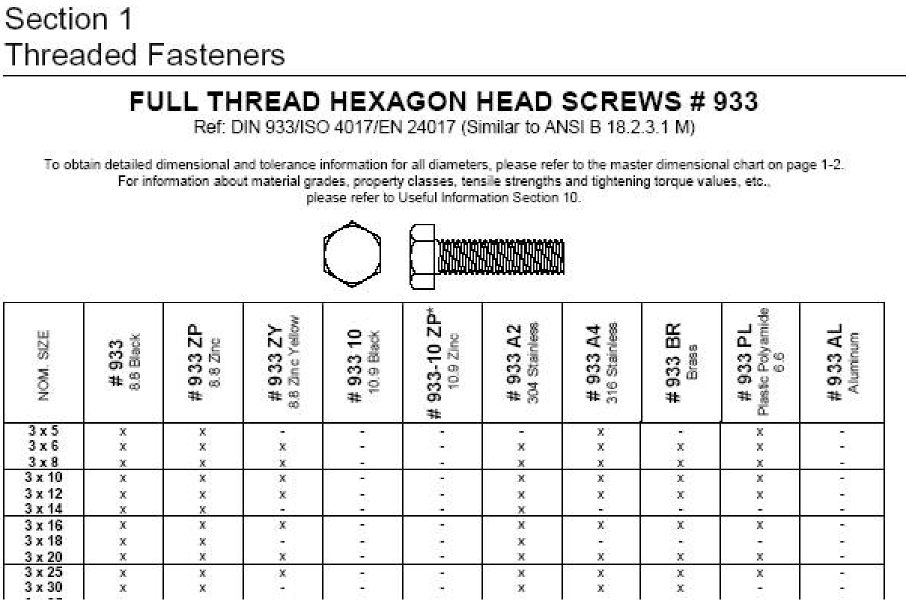


Figure 29 – Variables' specification in a fastener's catalogue for an ISO 4017 Hexagonal head bolt.



Section 1
Threaded Fasteners

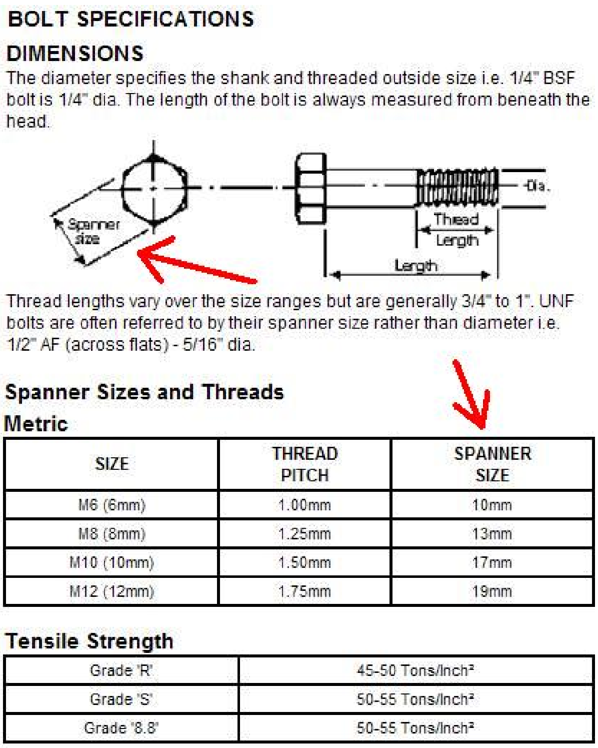
FULL THREAD HEXAGON HEAD SCREWS # 933
Ref: DIN 933/ISO 4017/EN 24017 (Similar to ANSI B 18.2.3.1 M)

To obtain detailed dimensional and tolerance information for all diameters, please refer to the master dimensional chart on page 1-2.
For information about material grades, property classes, tensile strengths and tightening torque values, etc., please refer to Useful Information Section 10.

NOM. SIZE	# 933 8.8 Black	# 933 ZP 8.8 Zinc	# 933 ZY 8.8 Zinc Yellow	# 933 10 10.9 Black	# 933-10 ZP* 10.9 Zinc	# 933 A2 304 Stainless	# 933 A4 316 Stainless	# 933 BR Brass	# 933 PL Plastic Polyamide 6.6	# 933 AL Aluminum
3 x 5	x	x	-	-	-	-	x	-	x	-
3 x 6	x	x	x	-	-	x	x	x	x	-
3 x 8	x	x	x	-	-	x	x	x	x	-
3 x 10	x	x	x	-	-	x	x	x	x	-
3 x 12	x	x	x	-	-	x	x	x	x	-
3 x 14	x	x	-	-	-	x	-	-	-	-
3 x 16	x	x	x	-	-	x	x	x	x	-
3 x 18	x	x	-	-	-	x	-	-	-	-
3 x 20	x	x	x	-	-	x	x	x	x	-
3 x 25	x	x	x	-	-	x	x	x	x	-
3 x 30	x	x	-	-	-	x	x	x	-	-

Figure 30 – Another supplier's catalogue page for the same component.

Despite these advisory international standard guidelines, very often the supplier's catalogue codification and properties list are delivered not compliant with the standard. Figure 29, Figure 30 and Figure 31 are snapshots from three different suppliers' catalogues showing the properties of an ISO 4017 hexagon head bolt.



BOLT SPECIFICATIONS

DIMENSIONS

The diameter specifies the shank and threaded outside size i.e. 1/4" BSF bolt is 1/4" dia. The length of the bolt is always measured from beneath the head.

Thread lengths vary over the size ranges but are generally 3/4" to 1". UNF bolts are often referred to by their spanner size rather than diameter i.e. 1/2" AF (across flats) - 5/16" dia.

Spanner Sizes and Threads

Metric

SIZE	THREAD PITCH	SPANNER SIZE
M6 (6mm)	1.00mm	10mm
M8 (8mm)	1.25mm	13mm
M10 (10mm)	1.50mm	17mm
M12 (12mm)	1.75mm	19mm

Tensile Strength

Grade 'R'	45-50 Tons/Inch²
Grade 'S'	50-55 Tons/Inch²
Grade '8.8'	50-55 Tons/Inch²

Figure 31 – Characteristics and variables definition in another catalogue for the same hexagonal head bolt.

Different terminology and classifications can be clearly identified referring to the same nominal parameters, and all divergent from the standard guidelines.

In these catalogues, the diameter of bolt's head is labeled by *flat width* (Figure 29) or *spanner size* (Figure 31), whereas the standardized variable designation is *s* (Figure 28). Figure 30 exemplifies yet another supplier's catalogue, where different table entries are adopted for the same bolt. This succinct real example, that is part of the daily PLC teams' modus-operandi, demonstrates the extension of the problem and the need for a methodology to contribute to solve it.

Due to the worldwide number of existing catalogue components and diversity of teams' culture, only in very specific situations one supplier would adopt the terminology and classification of a manufacturer or vice-versa. To impose a unique terminology and classification would be a solution. However, those suppliers not adhering to it most probably would be ignored, and this is not a favorable business situation, not permitting for instance an open selection of the supplier.

For that reason, to envisage forcing manufacturers or suppliers to adopt a specific ontology, even if it is based on the standards, does not work in most of the cases, especially when the involved organizations are SMEs. Thus, an advantageous solution would be to keep the terminology and classification in use by each one, and adopt a harmonized ontology to communicate between them. In this case, each team has to develop its own translator between its particular ontology and the harmonized one.

This development is to be done once, and without any expected difficulty thought they know in advance their own terminology and classification and the harmonized structure and inherent semantics. In the case of a required expansion, e.g., when the harmonized taxonomy supports a new supplier with extended properties, the respective translators would be updated accordingly to support them.

Therefore, the MENTOR methodology proposal of the present work addresses not only harmonization mismatching latent in already developed enterprise's ontologies, but bridges the current technology the company may have and the aimed technological shift [49]. The following section provides details of the proposal.

V.3. The MENTOR methodology

The MENTOR methodology is supported by an intelligent ontology system constructed as a hybrid system. Hybrid systems are a special class of knowledge representation systems which are constituted by two or more subsystems dealing with distinct portions of a knowledgebase and specific reasoning procedures [97]. Its aim is to combine the knowledge of different formalisms to improve representational adequacy and deductive power. In a certain way, MENTOR complements and details the harmonization methodology described in section *IV.3.3*.

A hybrid specialized ontology system needs to be able to interoperate with enterprise's proprietary ontologies. There are generally three approaches for combining such distributed heterogeneous ontologies [98]:

- Ontology Inclusion in which the source ontology is simply included within the target ontology;
- Ontology Merging using mediators;
- Ontology Mapping in which a part of the source ontology is related to the target ontology's entities.

Considering the listed options above, it was chosen the ontology merging and mapping as the more adequate approach to combine the engaged ontologies in the system, since the focus is to maintain enterprise ontologies and build a new one to be their reference in the domain. Therefore the described system is designed to facilitate semantic bridges between all the ontologies, which are related to ontology interoperability operations, where MENTOR provides an ontology (MO), ready to record the information related to all the ontology operations. The interoperability ontologies operations considered relevant are [59]:

- ontology mapping/matching, i.e. for each entity (concept, relation, attribute, etc.) in one ontology, a corresponding entity is defined in the second ontology, with the same or the closest intended meaning;
- ontology alignment, i.e. the process of bringing two or more ontologies into mutual agreement, making them consistent and coherent with one another [99];
- ontology translation, i.e. the process of changing an ontology representation language keeping its semantics unaffected;

- ontology transformation, i.e. the process that consists in modifying the structure or /and the properties of an ontology leaving unaltered its semantics;
- ontology merging/integrating, i.e. to build a new ontology starting from two or more existing ontologies with overlapping parts – merging, or extending some of existent parts - integrating;
- ontology checking, i.e. ontology information inconsistencies checking, this is commonly performed by reasoners or theorem provers [100];
- ontology evolution/versioning, i.e. ontology domain changes or adaptations to different tasks, over time.

The MENTOR - Methodology for Enterprise Reference Ontology Development is thought as a methodology that helps an organization to build and adapt a domain reference ontology.

Considering the listed methods in Table I, this methodology addresses all the referenced categories, i.e. ontology building from scratch; ontology reengineering; cooperative building and merge methods. Thus, MENTOR has a more comprehensive knowledge representation life cycle for the use in semantic interoperability existent problems, inside a domain business communications. It has been envisaged a tool based on the Protégé Java-based Application Programming Interface (API), to make operational MENTOR methodology functionalities. The developing work has served for corroborate the MENTOR's methodology steps described in the following sections.

MENTOR provides several step methods as semantic comparisons, basic lexicon establishment, mappings among ontologies and other operations on KB representations. This methodology is composed by two phases: the Lexicon Settlement (Phase 1) and the Reference Ontology Building (Phase 2) with three steps each (Figure 32).

In linguistics, the lexicon of a language is its vocabulary, including its words and expressions [101]. For a human, knowing a language implies having a mental lexicon, i.e. a memorized set of associations among sound sequences, their meanings and their syntactic privileges [102]. The Lexicon Settlement phase (Phase 1) represents a domain knowledge acquisition which comparatively to the human language apprenticing phase could be represented in computer science as a semantic organized structure with definitions.

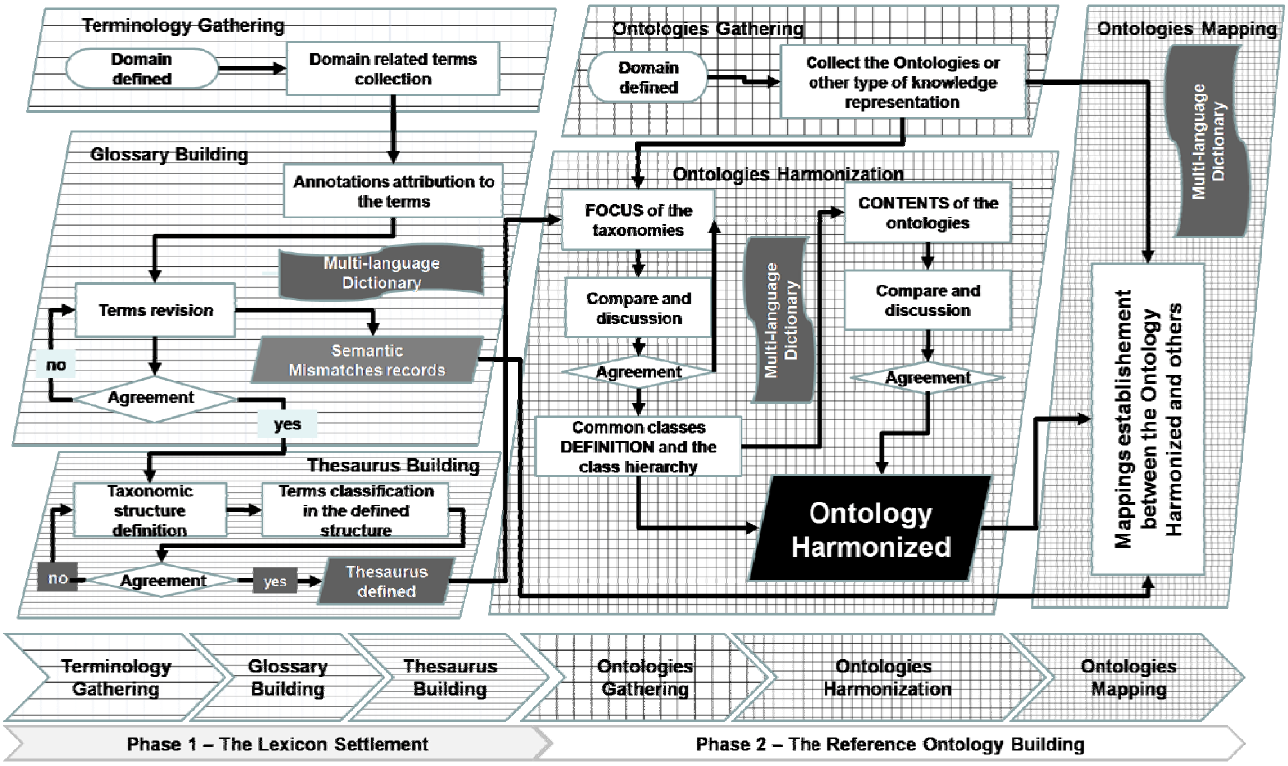


Figure 32 – Overall picture of MENTOR Phases and Steps.

The thesaurus can represent such words structure of associated meanings and thus should be built in order to establish the lexicon of a specific domain. This phase has three steps: Terminology Gathering; Glossary Building and Thesaurus Building. These steps were defined based on the UPON, which defines a set of workflows that establishes a thesaurus of the domain before starting the ontology building.

Figure 33 depicts the state diagram of the lexicon settlement phase. The terminology gathering step concerns the process of collecting all relevant terms in a specific domain previously defined. All the participants in the process should give their inputs. There is no rule from where the terms should come, since they are related with the domain established. Tools for automatic extraction of domain related terms can be found, as for instance, the *OntoLearn*. This tool aims the extraction of domain ontologies from web sites, and more generally from shared documents among the members of virtual organizations [103], nevertheless there is always need of a human checking before closing the terms list in order not to miss any domain terms. All the terms provided by the contributors are acceptable in this step. Nobody has authority to erase any other's participant term. The term should be collected with reference to the contributor in order each contributor provide term's annotation in the next step.

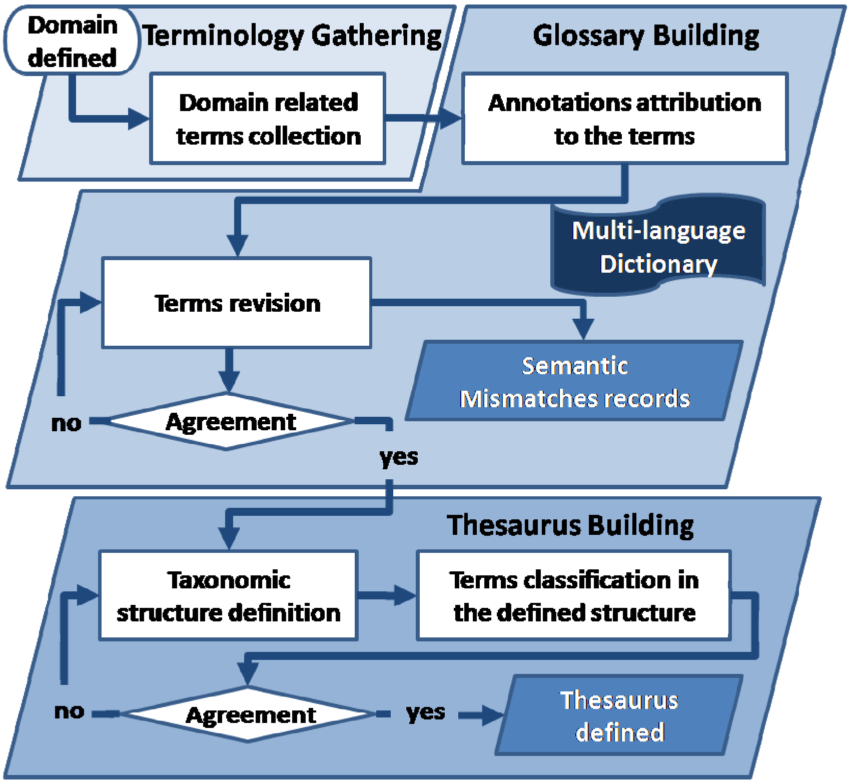


Figure 33 – Diagram of the Phase 1 – The Lexicon Settlement.

Glossary is a specialized vocabulary with corresponding annotations. This vocabulary includes terms that are unique to the subject and have special meaning in the field of interest. The annotations include descriptive comments and explanatory notes for the terms, such as definitions, synonyms, references, etc. A Glossary can be used when communicating information in order to unify knowledge sharing. The Glossary Building step intends to build a glossary in the domain defined. It starts with annotations attribution to the terms collected in the step before. Each contributor should provide the annotations for his own terms. After having all the terms provided with annotations, it proceeds to the terms revision cycle. In this cycle it could be useful to use a multi-language dictionary in case of the organization members don't use the same natural language. The dictionary will help translations to the agreed language for the reference ontology. The terms revision process can have four semantic and syntactic cases of mismatches:

- Two syntactical different terms with the same meaning description – the solution is to adopt one of the terms for being the reference in such semantics meaning. This process needs to be recorded has a semantic mismatch for future mappings;
- Two syntactical equal terms with the same meaning description – the solution is to erase one of them;

- Two syntactical different terms with two different meaning descriptions – no action needed, both must be kept;
- Two syntactical equal terms with two different meaning descriptions – the solution is to consolidate all the provided descriptions together in one of them and erase the other. In this last case, a new term could be proposed to the list if there is no agreement in the conjunction of the input descriptions and if the term to born is not present in the terminology list.

After a careful revision in all the terms with a successful agreement in their meaning consolidation, the glossary is defined from the terminology list in the domain specified. Another output from this process is the semantic mismatch records: this is made using a specific ontology described in the following section.

The Thesaurus Building step is composed by a cycle where firstly, the knowledge engineers define a taxonomic structure from the glossary terms, establishing some as thesaurus node terms. Secondly, the other terms are classified to the right paths in the existent taxonomic structure, being the thesaurus leafs. If there is an agreement in the structure and in the terms classified, the thesaurus is defined. If not, the cycle starts again. The thesaurus defined will enhance the ontology harmonization process in the next phase.

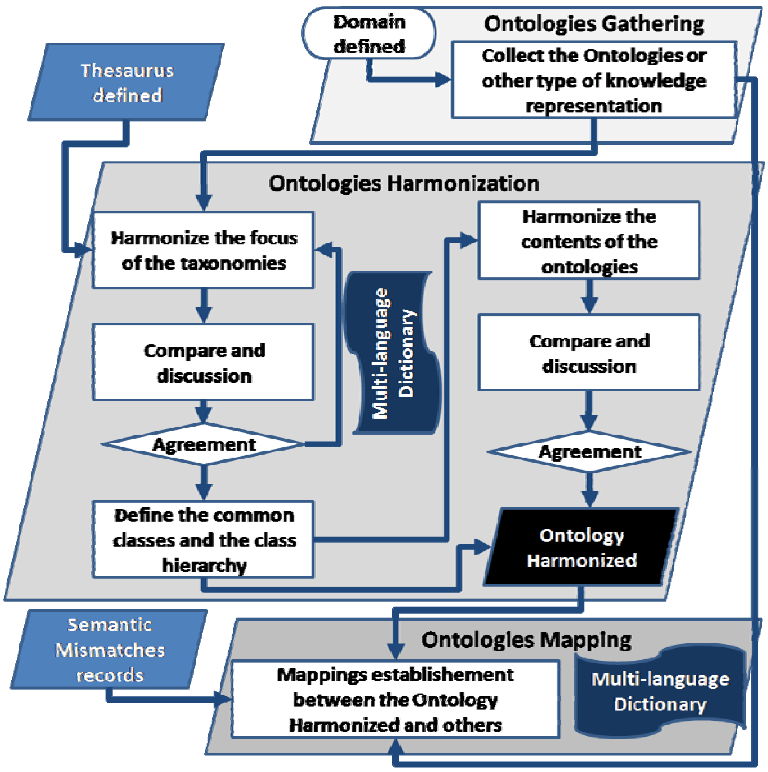


Figure 34 – Diagram of the Phase 2 – The Reference Ontology Building.

The Reference Ontology Building phase (Phase 2) is the phase where the reference ontology is built and the semantic mappings between the organizational ontologies and the reference one is established. Figure 34 describes its steps.

The first step comprehends ontologies gathering in the domain defined. Other type of knowledge representation could be used as input for the harmonization ontologies' process together with the thesaurus defined in the previous phase. The harmonization method for building ontologies defined by an adaptation made from Noy *et al* [83] [104], propose the development of a single harmonized Ontology's by two cycles where first the structure is discussed until having agreement on it and then the same process is followed for the ontology's contents definition. From this process new semantic conflicts could be found. After agreement, the resolution could be recorded[14] for further mapping establishments. With all the agreements accomplished, the harmonized ontology is finalized together with the mapping tables, describing the ontological relationships between the harmonized ontology and each one of the individual ontologies.

Semantic difficulties related to the natural language of the potential users of the harmonized ontology are likely to happen. To assist on it, the ontology is complemented with a multi-language dictionary where a set of normalized tokens gives the reference to the corresponding concepts and definitions in different native languages.

V.3.1. Mediator Ontology concept

Ontology mapping is an activity that attempts to relate the vocabulary of two ontologies that share the same domain of discourse [19]. The process of defining mappings between ontologies is not an easy task and requires a human support. MENTOR uses the *Mediator Ontology* (MO) as the reference for mediating the mapping establishment and its subsequent "mapping records" reasoning. One example is querying the MO for a correspondence to a reference term in a specific enterprise's ontology.

The MO is able to represent ontology semantic operations: the semantic mismatches found in the Glossary Building step; the semantic transformations identified in the harmonization process; the ontologies mapping; and other ontologies operations (e.g. versioning). MENTOR was built up as an extension to the Model Traceability Ontology defined by Sarraipa et al [105]. Traceability is the ability to chronologically interrelate the uniquely identifiable entities in a way that matters. The mapping relations can be related to

[14] The agreement will be recorded in the Mediator Ontology, as an instance. The concept will be explained in the following section.

a traceability element, in a sense that a specific term defined in the reference ontology has a related one in an organization's member ontology, considering ontologies as stages of the desired ontology life-cycle, that is, in this case, the reference ontology. This makes possible a way to trace ontology elements. The MO structure is represented in Figure 35 using UML class diagrams and Annex IV lists the MO expressed as an OWL ontology.

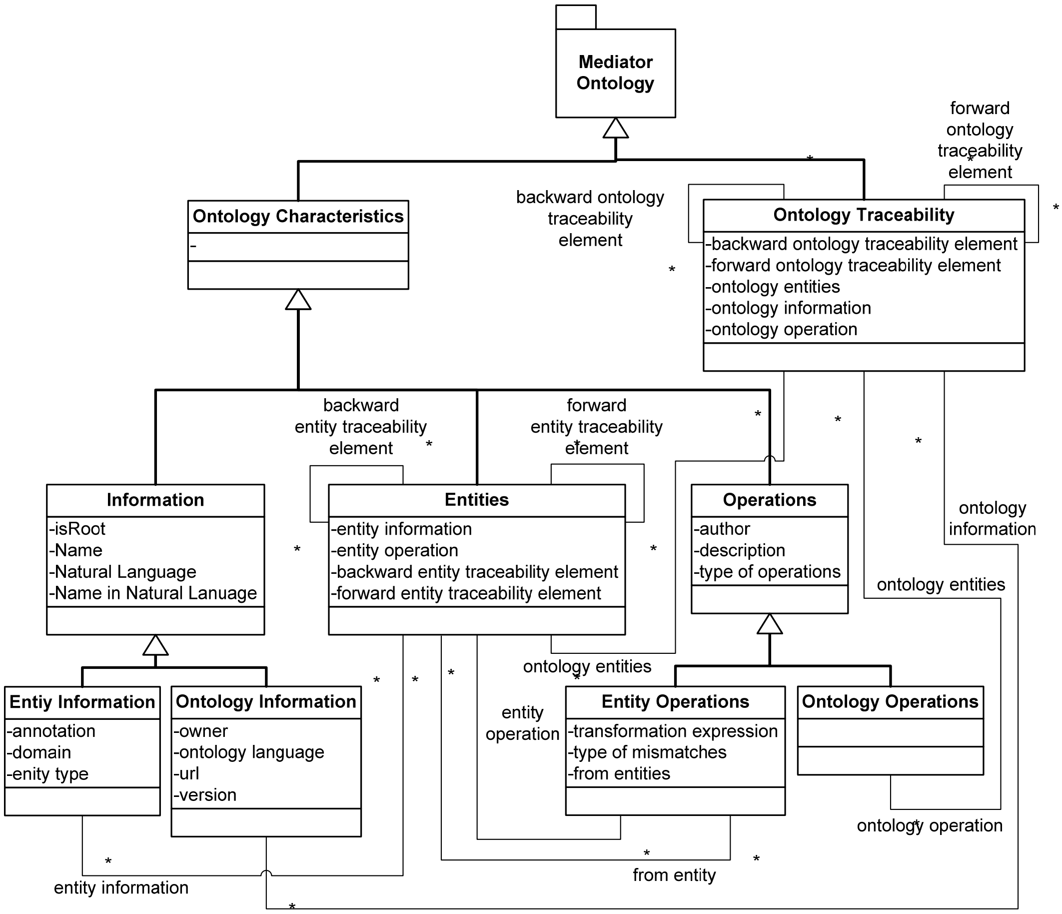


Figure 35 – Mediator Ontology structure.

The MO represents two classes: Ontology Characteristics and Ontology Traceability (see Figure 36). The Ontology Characteristics class represents:

1. general information ontology related to ontology and ontology entities (Classes: Information; Entity Information; and Ontology Information);
2. ontology operations that an ontology or an ontology entity (e.g. classes; properties; instances) suffered in the various stages of the ontology life cycle (Classes: Entities; Operations; Entity Operations; and Ontology Operations).

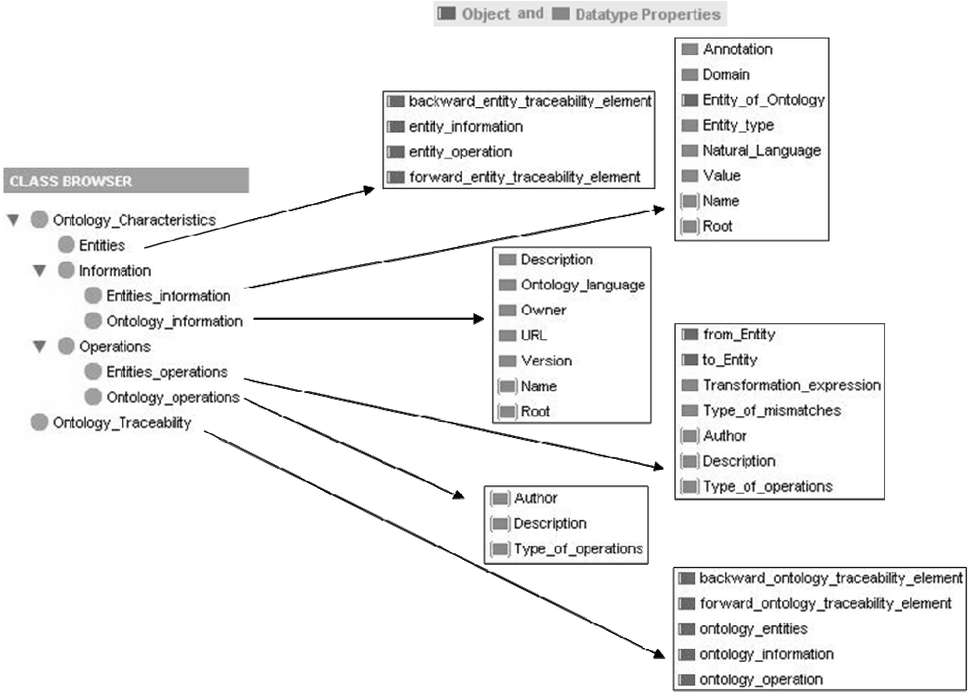


Figure 36 – Mediator Ontology overview (Protégé implemented).

The Ontology Traceability class represents the information related to the various ontology life-cycle stages. Figure 37 depicts an abstract ontology life-cycle example with three ontology operations and three stages. Ontology N is the intermediate stage.

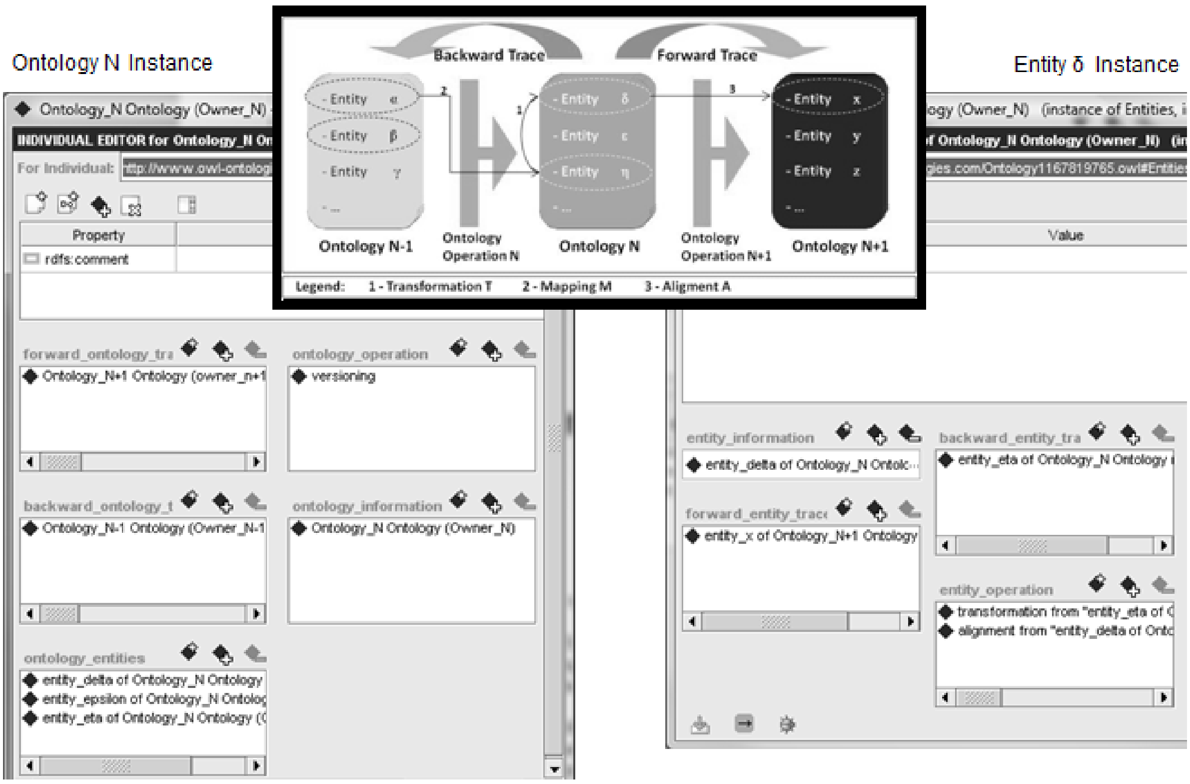


Figure 37 – Operations in a three stage ontology life-cycle.

From Ontology N is possible to make: a backward trace following an Ontology N entity to its related entity in the Ontology N-1 that is the root of this life-cycle, or a forward trace until the last version of the ontology (Ontology N+1). The Entity δ in the Ontology N is the result of a transformation that uses Entity η in the same ontology, which is related to the Entity α of the Ontology N-1. The Entity δ of Ontology N is in Alignment to the Entity x of the Ontology N+1.

In conclusion this ontology is able to log ontology and entity operations in a way that is possible to trace changes in all the ontology life cycle.

Since the objective of the MENTOR methodology is to build an ontology that represents the knowledge of an organization domain, but keep working the old own enterprises ontologies' developments, it is important to define how or in what processes all the ontologies involved should be maintained.

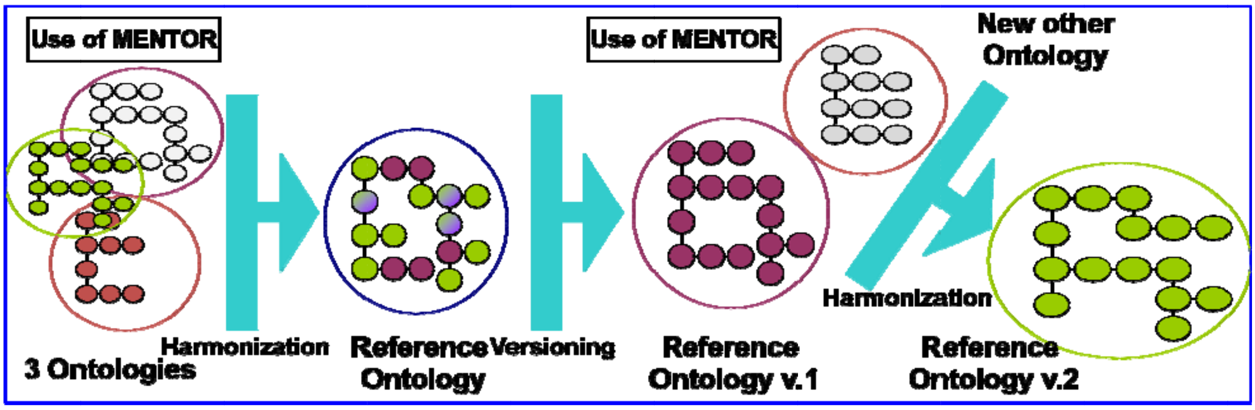


Figure 38 – Ontology Maintenance using MENTOR

To avoid inconsistencies in the MO a collaborative method must be defined for updating it when a related ontology is changed. Another important input is when new enterprises want to interoperate with an existent MENTOR compliant organization. The MO supports such integration that can be represented as a new stage in the reference ontology life. The picture in Figure 38 shows a three stages instance, the last referred issue being represented.

Reference ontologies must aggregate all the enterprise members' knowledge. Adding new enterprises to an existent organization implies to integrate all the new enterprises knowledge in the organization reference ontology. For this, new enterprises knowledge must follow all the MENTOR phases. In this case there is already a glossary, thesaurus and a reference ontology established. Thus, the process is to follow the entire MENTOR steps taking in account the results from the first building reference ontology, consequently all the steps will have a lighter discussion or process in their outputs. Only a slight refinement will be needed, since the previous results have a bigger weight in the new reference ontology version consolidation.

An illustrative example making use of MENTOR is included in the next section *V.4 – Ontologies harmonization to assist Manufacturing*, where further details are provided.

V.4. Ontologies harmonization to assist Manufacturing

V.4.1. Scenario experimented

An industrial scenario was put up to address the interoperability problem between manufacturers and suppliers of components, when they face communication difficulties generally resulting from the lack of knowledge regarding data representation and semantics of the other interlocutor.

Many characteristic situations were identified where this problem exists along the manufacture chain and product life cycle. For example, whenever it is needed to search for information about a component during product design or maintenance, and the potential suppliers do not use the same catalogue's designation or classification.

Remarkably, although many of these organizations already use information technology to support the supply chain management, in the current manufacturing environments and particularly in the SME segment, the search in catalogues and exchange of information is mostly paper based, and the described interoperability problem is well recognized even at a primary level, where the interlocutors are both humans.

Figure 39 illustrates the scenario, describing phase by phase one of the typical situations where furniture manufacturers find difficulties to select and obtain specific mechanical components for design and production. For simplicity of reference in the text, the phases (1-5) described are numbered and identified in the figure. Also, the language adopted to explain it is the one in practice by the industry.

The first phase starts when the design team demands to search for a specific mechanical hexagonal bolt. The required characteristics of the component are communicated to the procurement department (phase 2), responsible to identify and query candidate suppliers. The search is made by a mediator consulting the catalogues available in house, and making phone calls or sending a fax to directly contact the suppliers.

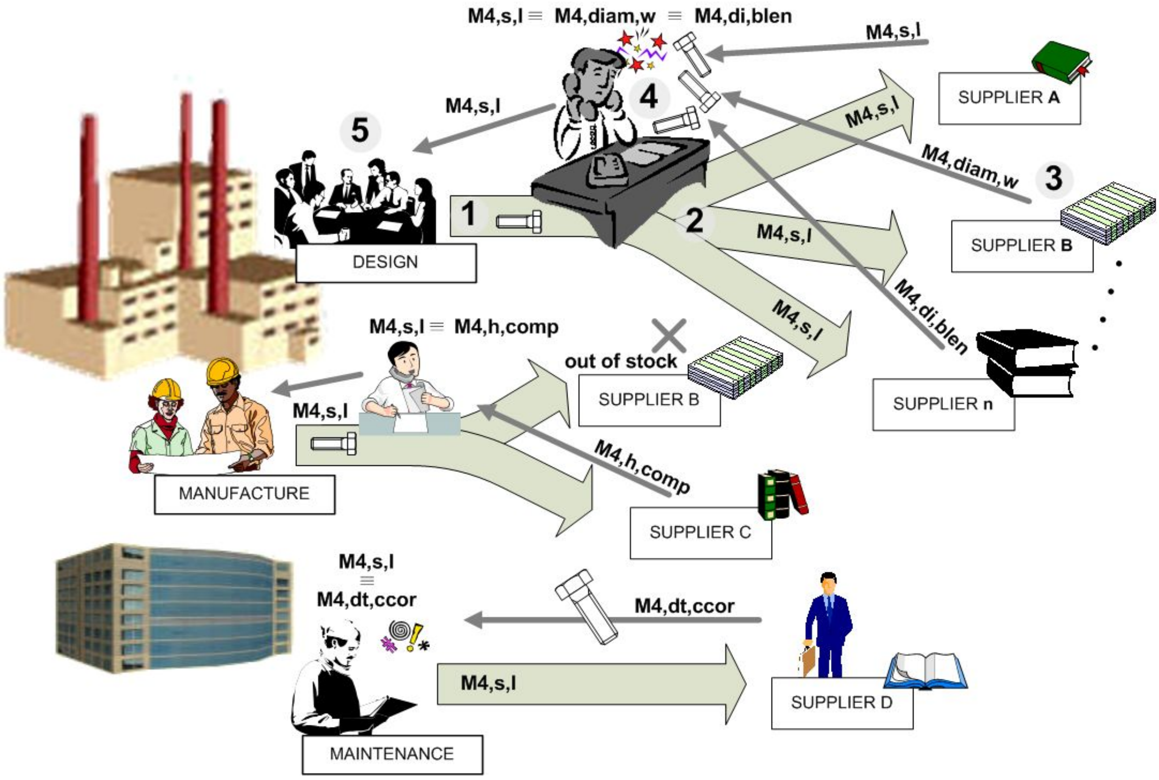


Figure 39 – The scenario experimented.

Suppliers respond (phase 3) verbally or sending their leaflets and catalogues with information about the component. However, each one is using different nomenclature and variables names (most proprietary) to describe the same physical component. Thus, a dialog with the supplier is started, to validate the compliance between required properties and characteristics of candidate parts, and clarify definitions and redundancies between nomenclatures.

Next phase (4 in the picture) is a manual job and a mental challenge, where it is necessary to identify and establish the correspondence, i.e. mapping, between the different supplier's nomenclatures and the one in use by the design team. Then, the component's reference code is delivered to the design team (phase 5).

In addition, Figure 39 illustrates two other typical situations where similar problems occur: one involving the manufacturing department, and the other the maintenance team. In these two situations, the chosen supplier runs out of stock or founds discontinued the searching component. Consequently, both situations require a new try to find another equivalent component, following the searching procedure as described before.

The introduction of the proposed methodology in this scenario was recognized advantageous by the participating users. When the method was applied to combine a set of selected catalogues of components, it resulted in a harmonized ontology defined along with the mapping table describing the relationships between the manufacturer's component classification and those defined by each supplier's.

Using this result, the mediator can manage the harmonized ontology together with the table with the mapping rules between the manufacturer's and the suppliers' schemes to describe the components in search. This is of valuable help to the mediator, because in a glance he can have a systematic and immediate understanding about the components he is managing, enabling him to actuate faster and more accurately.

Over time, the interoperability under this scenario is improved whenever more and more catalogues are combined by the manufacturer and the respective mapping table updated. Nevertheless, if for any reason the harmonization of a new catalogue does not take place, the components described in it can still be chosen. However, in this case the mediator can not take advantage of the available mapping guidelines, and should proceed as habitually did before.

Figure 40 illustrates the framework for the use of the harmonized ontology and mapping table in the context of the experimented scenario.

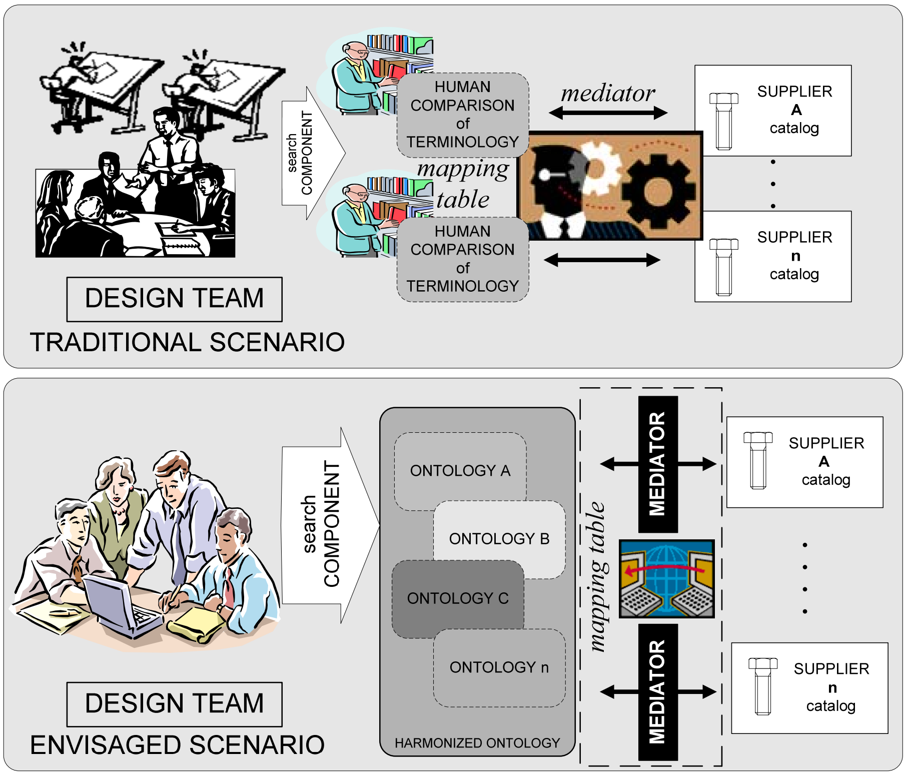


Figure 40 – Framework for the use of the harmonized ontology and mapping table.

This Figure 40 is divided in two parts. The upper side depicts the use of the framework when applied to the manufacturing environment as it was found when the study started. However, in the advent of the internet and networked organizations opportunities, these organizations aim to start using more and more computational systems to manage these activities, and the web to communicate between them.

In this envisaged scenario, the harmonized ontology and mapping table are materialized in an open-source software platform, Protégé. With it, the mediator is assisted getting a transparent interface between the information from the users (e.g., manufacturer's design team) and suppliers' catalogues.

In this moment, the motivation is to extend the envisaged prototype, and move towards a complete automatic scenario where the search and selection of components would be directly executed by the involved parties through a *computational agent*, completely automated and embedded in a web application.

V.4.2. Using MENTOR methodology

The simple choice of a "bolt" supplier by a mechanical engineer/designer, very often brings interoperability problems. Suppliers have defined various nomenclatures for their products and its associated knowledge. As referenced before, even after the appearance of standards in bolt specifications the problem persists. Thus, the need to align product data and knowledge emerged as a priority to solve the dilemma.

The presented problem was used as a MENTOR use case scenario for verification and illustration purposes. The work starts with a reference ontology building related to an organization composed by two "bolt" suppliers, and then is followed by presenting a MO application for semantic messages translation.

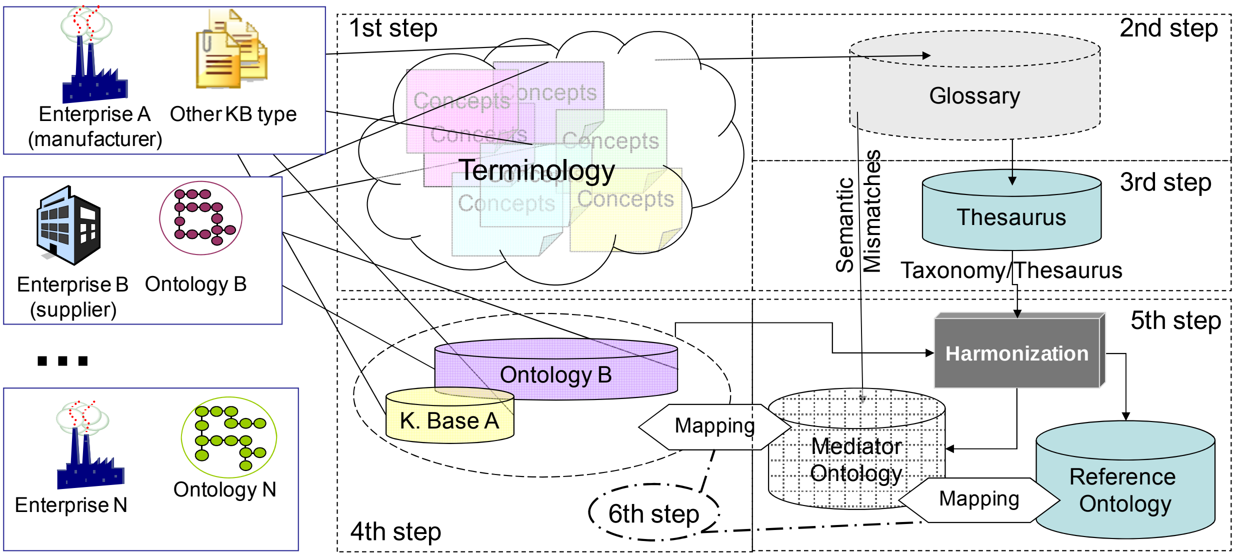


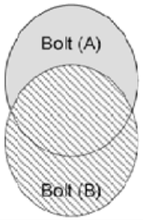
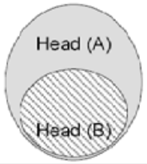
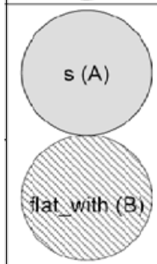
Figure 41 – MENTOR use case scenario.

Figure 41 pictures the validating scenario, where two enterprises agreed to build a reference ontology to be their knowledge front-end to their clients, though there where the condition to maintain their own meaning and nomenclature for products of each other. Although from one side one of the enterprises has its product data represented through an ontology, the other still present traditional product catalogues - regardless the electronic version they merely were digital versions of catalogue pages. The methodology used for this reference ontology building was the MENTOR methodology.

The six main steps identified in Figure 41 represent the ones identified by the MENTOR methodology. The two first steps (Terminology Gathering and Glossary Building) are summarized in Table III. The final reference ontology of Fasteners was included in Annex II for reader's assessment purposes.

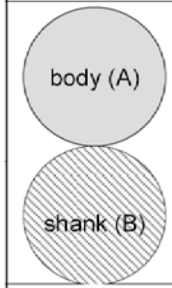
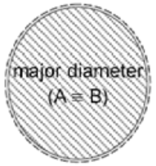
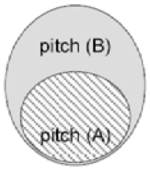
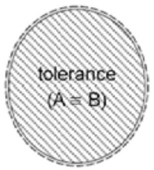
The domain and experts engineers started to collect their own terminologies together, resumed their respective own terms and definitions. The "Term" column represents the Terminology Gathering step of the two involved enterprises. Both lists of terms are placed side-by-side with their respective explicit definitions as they are considered in each enterprise. Although this could seem a simple task to complete, in some electronic catalogues, neither there are term definitions at all, nor explicit meaning, scope or usage of them with reference to the product or product model. Moreover, usually standard references are used to replace all the underlying concepts and definitions.

TABLE III – Fastener Terminology Gathering and Glossary Building example

Variables and definition matching	Original ontologies			Reference Ontology		
	Term	Category	Definition	Term adopted	Category adopted	Reference definition
	Bolt	class	headed fasteners having external threads that meet an exacting, uniform bolt thread specification (such as M, MJ, UN, UNR, and UNJ) such that they can accept a nontapered nut	Bolt	class	headed fasteners having external threads that meet an exacting, uniform bolt thread specification (such as M, MJ, UN, UNR, and UNJ) such that they can accept a nontapered nut
	Bolt	class	term used for a threaded fastener, with a head, designed to be used in conjunction with a nut			
	Head	class	end part of the fastener, which is used to support the driving device of the fastener	Head	class	end part of the fastener (considered as upper part), which is used to support the driving device of the fastener
	Head	class	upper part of the fastener that is used to turn around the fastener			
	s	property	dimension across flats in a hexagonal head	nominal diameter	property	in an hexagonal bolt's head, is the dimension of the nominal diameter tangent to the flats (also expressed as the dimension across flats which corresponds to the size of wrench to use)
	flat_with	property	diameter across the flats of the bolt's head. It is also the size of wrench to use.			

(continues next page)

TABLE III (continuation) – Fastener Terminology Gathering and Glossary Building example

Variables and definition matching	Original ontologies			Reference Ontology		
	Term	Category	Definition	Term adopted	Category adopted	Reference definition
	body	class	the part of the fastener that support the screw thread	shank	class	the part of the fastener that support the screw thread and necessary extension of unmachined surface
	shank	class	lower part of the component with the thread surface and necessary extension of unmachined surface			
	major diameter	property	the diameter of an imaginary cylindrical surface tangent to the crests of an external and(or) to the roots of an internal thread	nominal diameter	property	the diameter of an imaginary cylindrical surface tangent to the crests of an external and(or) to the roots of an internal thread
	major diameter	property	diameter of an imaginary cylinder parallel with the crests of the thread; in other words it is the distance from crest to crest for an external thread, or root to root for an internal thread.			
	"*passo*" translated to "pitch"	property	the axial distance between a point on a thread flank and the equivalent point on the immediately adjacent and corresponding flank	pitch	property	The axial distance between a point on a thread flank and the equivalent point on the immediately adjacent and corresponding flank. This property enables the interconnection with complementary surfaces
	pitch	property	distance between similar points on adjacent threads, property of a thread that distinguishes and enables the interconnection with complementar surfaces			
	tolerance	property	Permissible limits of variation that a measure can fall within; determined by the inspection phase after manufacture of the component	tolerance	class	interval of values of allowable deviation from a nominal or specified dimension
	tolerance	property	Allowable deviation from a nominal or specified dimension, determining maximum and minimum material condition			

The diagrams in the left column represent the human interpretation of such definition and represent the terms as well. After comparison, discussion and agreement of what meanings and terms should be adopted, the terms to be used in the glossary are condensed in the righter part of the table. The diagrams were chosen to show the variety and different relationships of term-definition sets that are possible to found.

At this stage (of Glossary consensus), concerns about the structure and correspondence of the semantic concepts are not under consideration. Experts involved must have this in mind during this stage or otherwise the exercise may become endless and quickly ruin the reach of consensus. Thus, a simple side by side comparison of the terms and their definition together with the aid of diagrammatic interpretation will simplify and clarify the Glossary reference building. Summarizing, it may be considered conjunction, inclusion, disjunction and coincidence of the terms' definitions, after the diagram elucidative representation, and with the overall agreement this corresponding 2nd step could be considered finished.

Additionally, an example is included where the multi-language dictionary demonstrates its usefulness and relevance. Concepts that may exist in other language than the common one would be translated and the process will proceed smoothly under the reference language (see information related to the term "pitch" in the Table III). This detail assumes particular relevance when considering the broad diversity of electronic catalogues' editions considering different languages. Additionally, when a local customer uses the web, dissimilar terms expressed in a different language are a real obstacle to ensure the correspondence of product properties and real components matching, and conversely to do business in a successful manner.

By assisting the last referred step (Glossary Building), the MO - according to the principle of traceability described earlier - logs the necessary steps and options that the second phase (The Reference Ontology Building) will handle.

While the main purpose of the second step is to uniform and construct the reference platform of terms and definitions, it is relegated to the next step - the Thesaurus building. In this step, the relations of concepts are identified: previous connections and existent relationships are considered among the glossary terms and respective definitions. As described earlier, possible hierarchy of concepts in the reference ontology may need to be re-arranged, despite the previous individual arrangement that each enterprise may have found and adopted. This Thesaurus building step is responsible to clearly identify the relationships that terms have between them, their nature and extension. As the diagram of

Figure 41 shows, Thesaurus consensus points to harmonization process: its contribution will mainly feed the Reference Ontology Building.

One example of such steps is related to definition of a reference concept that was defined from the two following proprietary concepts:

• **Enterprise A** -> Concept: "s"; Definition: "dimension across flats in a hexagonal head";

• **Enterprise B** -> Concept: "flat with"; Definition: "diameter across the flats of the bolt's head. It is also the size of wrench to use".

That resulted in the following reference concept:

• **Reference** -> Concept: "major diameter"; Definition: "in a hexagonal bolt's head, is the dimension of the nominal diameter tangent to the flats (also expressed as the dimension across flats which correspond to the size of wrench to use)".

These semantic mismatches were recorded in the MO, in order to be used in further mapping establishments.

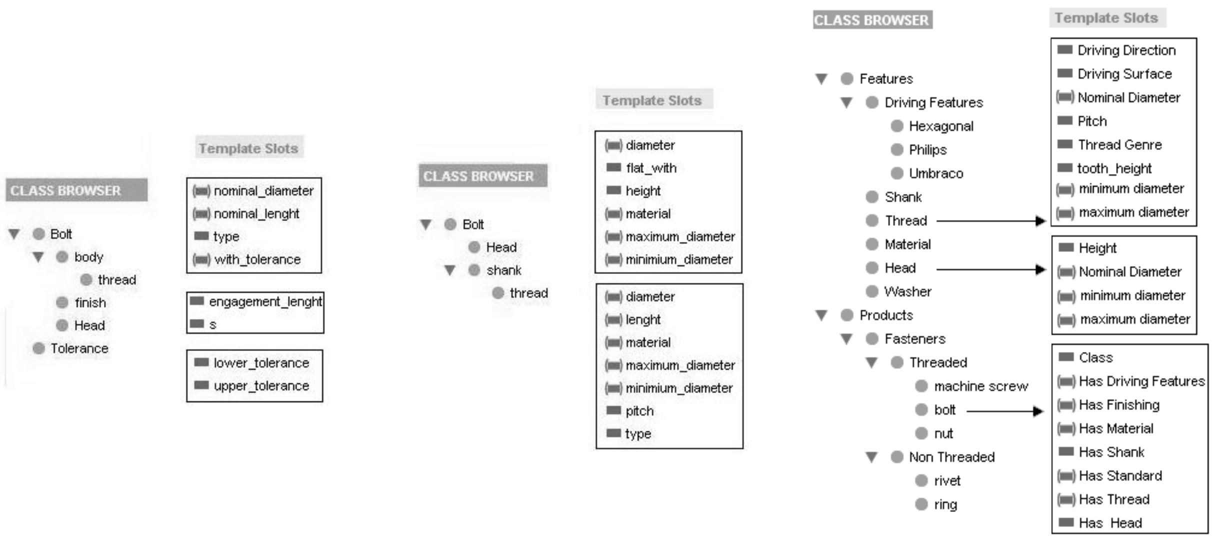
The 4th step Ontologies Gathering is made by the simple process of collecting the knowledge of these two involved enterprises.

In the step 5, the enterprise ontologies together with the thesaurus are harmonized in a new ontology – the reference Ontology (Figure 42). This process is assisted by the logs existent in the MO. The knowledge engineers are able to see semantic bridges between enterprises own terms with the ones established and present in the thesaurus. After that, they could see the best ontology structures that they should use. This reference ontology is able to represent any kind of bolt. Figure 42 highlights the classes *Thread*, *Head* and *bolt* that are related by the properties *Has Thread* and *Has Head*. This kind of relation is represented in the thesaurus as a direct hierarchy, although it is during the harmonization that properties are defined (*Has Thread* and *Has Head*) which should relate the mentioned classes.

Sometimes during the harmonization phase previous concepts are semantically redefined. In this scenario, one semantic issue emerged, related to the concept "*major diameter*" (mentioned above - check Table III). In the Glossary definition step, both enterprises' engineers totally agreed in one definition for the "*major diameter*" concept, taking in consideration its relation to the *head* class domain and by the conjunction of the *s* and *flat width* concepts proprietary definitions. Nevertheless, in the reference ontology, this concept was replaced by "*nominal diameter*" that was already defined in the glossary for the *thread* class domain. Conclusion: the concept "*major diameter*" was rejected and appeared the "*nominal diameter*" concept as the one to be related to both domains *thread* and *head*. The "*nominal diameter*" reference definition is presented in the following, where the first definition was the first one defined and which only took in consideration the thread class domain. The second is the final version, which the engineers tried to adequate to both classes' domain (*thread* and *head*):

- **First Reference** -> Concept: "*nominal diameter*"; Definition: "the diameter of an imaginary cylindrical surface tangent to the crests of an external and (or) to the roots of an internal thread".

- **FINAL Reference** -> Concept: "*nominal diameter*"; Definition: "numeric value used to quantify the diameter of a cylindrical surface envelope of the feature, without precision neither tolerances considerations (either geometric or dimensional).



Ontology A (Retailer) Ontology B (Manufacturer) Reference Ontology

Figure 42 – Snapshots of the proprietary Enterprises and Reference Ontologies.

Table II only shows a small part of the terms used by these enterprises concerning the bolt domain.

Ahead another example is presented, which intends to present the usefulness of the MO in supporting transformations of data. Engineers defined two properties, *maximum diameter* and *minimum diameter* as a reference related to the tolerance characteristic of a bolt. Moreover is indicated that tolerance was defined as the interval of values of allowable deviation from a nominal or specified dimension. These properties are equally used by the enterprise B. But, Enterprise A, uses the concepts *upper tolerance* and *lower tolerance* referring to such properties, which represents the same expected result but using different data values. Thus, it was needed to establish a transformation expression to relate them. Since *nominal diameter* concept has the same value and semantics in all the ontologies, from Reference to A ontology, the transformations equations related to the tolerance properties are the following:

upper tolerance = *maximum diameter* - *Nominal Diameter* (1)

lower tolerance = *Nominal Diameter* - *minimum diameter* (2)

And, from the A to Reference ontology the transformations equations are:

maximum diameter = *nominal diameter* + *upper tolerance* (3)

minimum diameter = *nominal diameter* - *lower tolerance* (4)

Since all the ontologies operations (mappings and transformations) are saved in the MO, appropriate queries could be used for semantic translations between the organizations members, including a hypothetical organizational front-end which uses the established reference knowledge. The example of Figure 43 illustrates what happens if a customer wants to buy to an organization one specific bolt. The client system sends a "*getProduct*" message of a bolt, in which the thread has a *nominal diameter* with a value equal to 10; a *maximum diameter* of 10.2 and a *minimum diameter* of 9.9. Then, the system's Mediator translates the message and forwards it to the bolt suppliers. Finally, each of them receives the message with their recognized semantics and data.

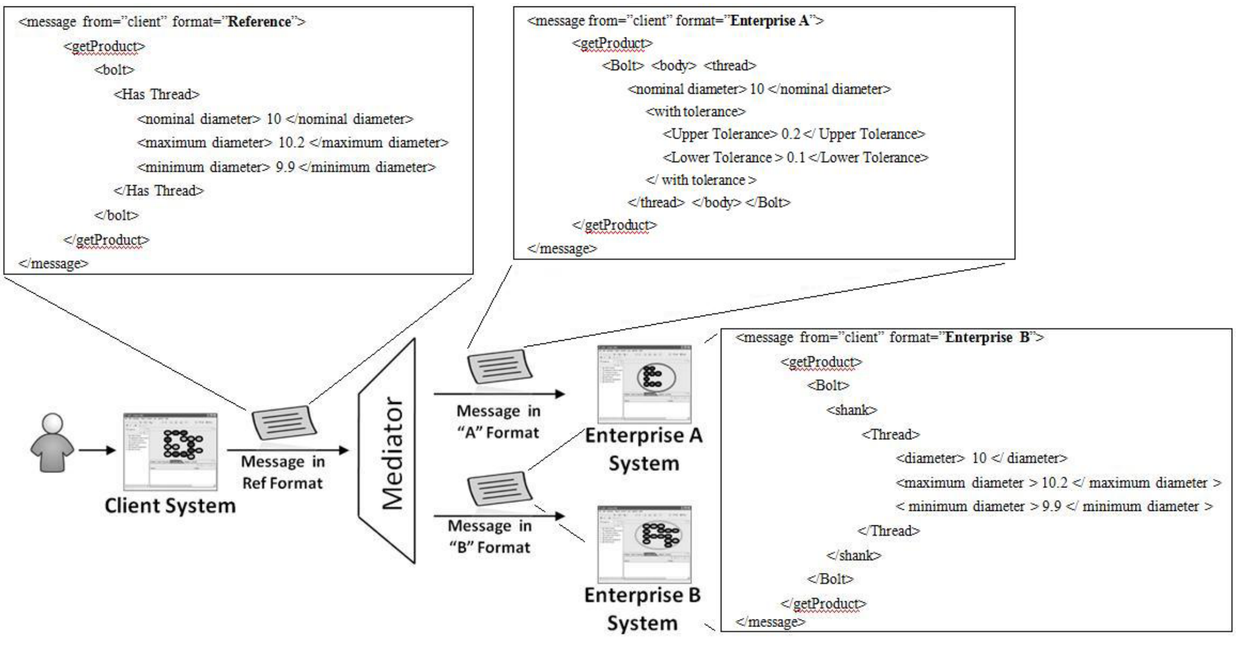


Figure 43 – Mediator's Message Translation Example.

The proposed MENTOR methodology enhances inter and intra-organizational knowledge sharing, allowing its actors keeping their own ontologies or knowledge representations and contributing to a reference ontology in the domain. MENTOR brings together the building and reengineering of ontologies related to mapping competences.

Ontology maintenance is another characteristic that MENTOR facilitates by enabling traceability recording in MO and which information could be used to track changes or to go back to consistent previous ontology versions. MENTOR also enables dynamic and flexible seamless joining of enterprises to develop business in a network of partner organizations.

Several advantages resulted through the MO use during communication among client and suppliers: a short term advantage was the acquired autonomy of computational systems of any enterprise to smoothly communicate with external parties as they were using the Reference ontology (which latent knowledge richness likely offers new business opportunities). This is also the main motivation that Enterprises may consider to join the Reference ontology building process, independently of its domain expertise or budget impact in the market.

Medium and long term advantages of described MO methodology adoption are also expected after the described encouraging results. In fact, the MO methodology introduces enhancements to the very early stages of product design and development, though it is expected that semantic correlated with product data models were an added value during product’s manufacturing phase - not only at data level mismatches but also in manufacturing specifications assessment too. Lastly, in recent years, parallel efforts of research community lead to the development of computational Product Models and Product Data Models, like those resulting from ISO 10303 STEP technology (addressing engineering issues) and more recently those from Semantic web (focused on business aspects). Both technologies have a strong potential in their specific application range, but promising enhancements may arise when providing existent product data models with semantic capabilities, thus merging those two worlds. The STEP computational product model offers a manufacturer the necessary confidence to manipulate geometric product data, but extending knowledge capabilities of such model (and in particular semantic enrichment) is a key improvement to fully profit of emerging electronic commerce.

MENTOR has been prototyped (Figure 44) with the main part of the functionalities described in the methodology proposed above. In the right part of the Figure 44 is shown an interaction with a domain dictionary (e.g. furniture), which helps in some definitions and by natural language translations. The methodology has been tested by the funStep initiative under the INNOVAFUN project (www.funstep.org) in their furniture reference ontology building. The thesaurus and the reference ontology built in such process have been used for testing and consolidation of the ISO 10303-236 (Product data representation and exchange standard) model semantic enrichment.

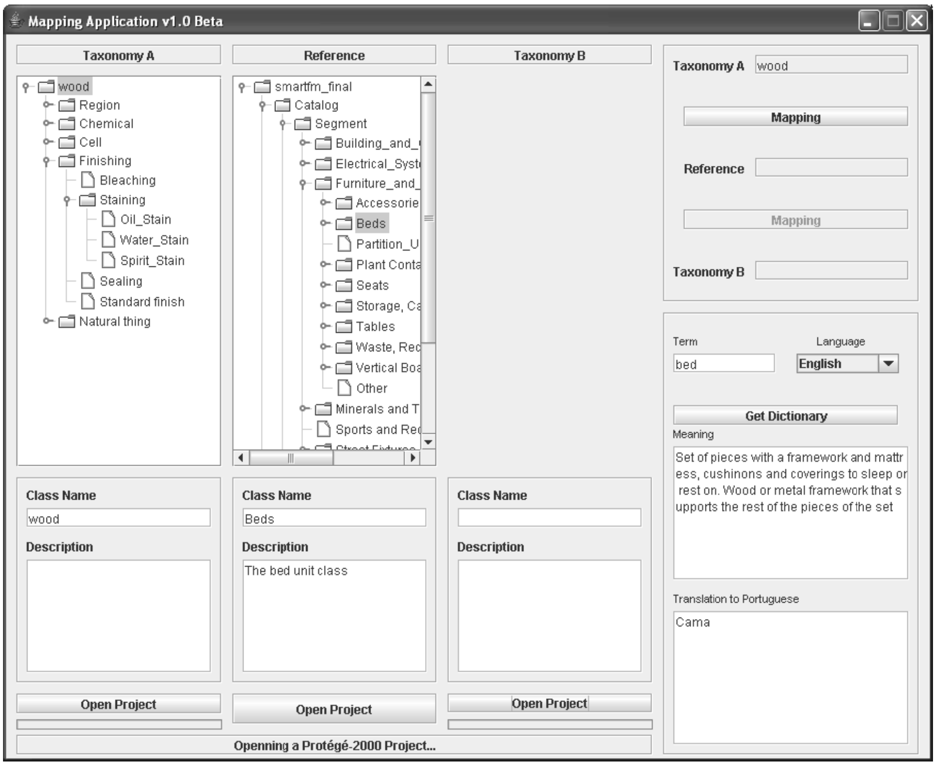


Figure 44 – MENTOR Prototype Tool.

Protégé is a well known and widely recognized ontology and knowledge-base editor developed at Stanford University. It is an open-source Java tool that provides an extensible architecture for the creation of customized knowledge-based applications [106].

The platform developed (Figure 45) recurred to Java technology to extend the Protégé editor, and adopted a recognized open mechanism for implementation of web-enabled services [107].

Based on the consolidation of the state-of-the-art in platforms for ontology development and management, the architecture of the platform described in Figure 45 is envisaged to assist the development and usage of the harmonized ontologies through the web.

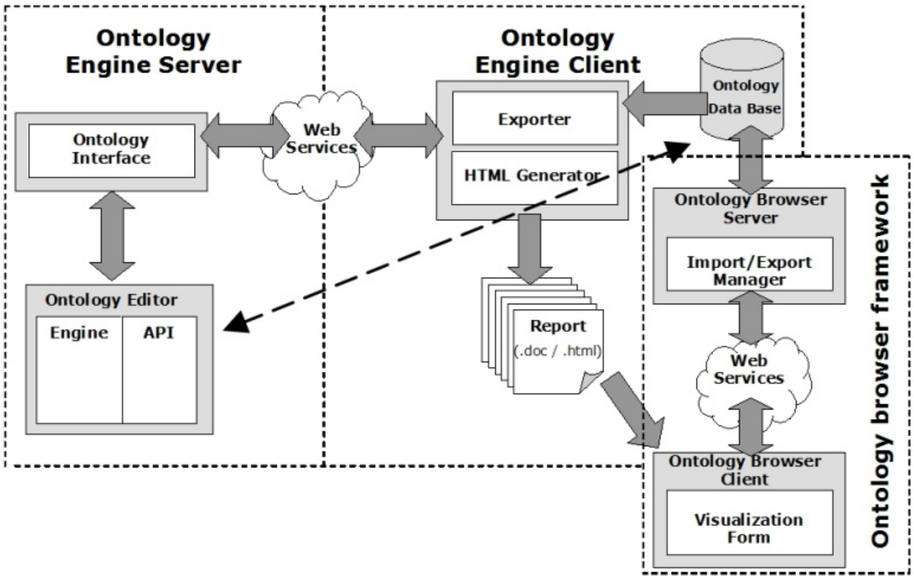


Figure 45 – Architecture envision of the platform to extend MENTOR capabilities.

MENTOR prototype and *webservices* architecture are the operational computational parts to replace the human based scenario outlined in Figure 39. Basically, each PLC agent needs to ensure an intelligible communication with the peers, either adopting a reference ontology or developing its own. An up-to-date Mediator Ontology will assist the PLC agent in the data interpretation (the agent “views” external data as its own ontological entities and structures) to ensure the message content understanding.

It makes no difference if the nature of PLC agents herein referred is inside or outside organization structure, since under multi-agents technology and Enterprise Models scenarios it is expected they may be interoperable, acting automatically and intelligently on their own.

Chapter VI

VI. Conclusions and Future Research Trends

Within the globally scaled economy and markets, the production process is re-emerging as the value-creator activity and the main source of enterprise revenues. The worldwide growing market contributes to the increase of customers' exigencies, in terms of both quality and delivery times. The product itself is becoming more complex, as a combination of physical components and services. The Product Life Cycle concept emerged, which required the explicit integration of costumer requisites and all other specifications of PLC activities into the final Product. The Final Product characteristics are the result of all PLC activities, and the best way to get some proficiency is by re-using previous data and knowledge.

The availability of a computational product representation, where geometric and technical product data may be incorporated makes possible the integration of services and applications among multiple production phases and the myriad of computational tools of PLC.

The particular progress made through developments in the STEP standard Product Data Model has had a proficient impact in industry competitiveness. But, if initial data and information drawbacks of Product Models were suitably addressed by STEP and to some extent solved (e.g. with parameterization), knowledge skills and other integration properties started to be required as a key requisite.

The production process conceptualization is changing, starting from market understanding, through product and process design, to operations and distribution management, often exceeding the boundaries of a single organization.

The formation of cooperation and collaboration alliances between several small organizations is proving, in multiple cases, to be more efficient and competitive by comparison with big companies. This is typically what leads companies to join efforts to survive in very evolutionary and dynamic markets. Moreover, Enterprise Modeling strongly contributed to generate new concepts like Extended and Virtual Enterprises, which impact goes beyond singular organizations work.

However, partnerships cause some problems, mainly in integrating Product Life Cycle phases, since manufacturers, distributors, designers, retailers, warehouses, often use their proprietary solutions which are, typically, not interoperable with one another. As in manufacturing, the necessary incorporation of different brought-in parts requires detailed data check and update by different teams, using heterogeneous applications to keep valid the initial design conditions and maintain the product assembly consistency. This task is often facilitated with computational tools, and the components' manufacture specifications should be of common understanding, to reduce re-work and time consuming during catalogue examination.

The information managed by one organization habitually embraces situations of broad heterogeneity, where concepts need to be handled under different structures, knowledge modeling components and levels of access and detail. It results that, in general, the information managed in each PLC stage regards to different classifications, utilization scope and teammates' relevance.

Coordination between several independent teams is the key factor to stay competitive in growing global market, and if data exchange assumes crucial role in enterprise efficiency more critical is its usefulness and understanding.

For an enterprise to achieve seamless information interoperability, a common agreement of the global information system's structure and semantics is required. Only in this way will it be possible to unambiguously describe, internally and externally to the organization, the complete meaning of the data instantiated and exchanged through the enterprise models.

This work proposes a methodology which combines many ontological worlds in place, i.e., the instances of domain, developing a harmonized ontology aiming to represent a domain of discourse. To obtain this consensual model, it is necessary to classify and combine the concepts from the different sources within the domain of applicability, describing them in a unique harmonized hierarchal structure of classes and definitions. This situation of common understanding enforces the business relationships between organizations, and facilitates the internal communication between the different organization's departments and services.

The proposed MENTOR methodology contributes to a gradual replacement of human variables matching (an actual time-spending and bothering commitment) by smooth computer to computer communication, constituting a skillful option. Moreover, collecting basic knowledge (like those of mechanical elementary shapes) through both the capture of product and process restrictions, increases the potential of existent knowledge repositories, which comprise by know an accurate manufacture specifications support. Hence, such knowledge repository may grow based on capture of simple rules, which, being processed by first order logic applications, clearly identifies troubles or incompatibilities between designed components and particular resources of each supply chain.

Several advantages were identified, in result from the use of MENTOR methodology, the most relevant being the semantic enrichment of standard product data models developed under ISO 10303 STEP standards. Product data models are well defined through STEP models, with necessary and sufficient geometric 3D detail, but the lack of expressivity of such models was been identified as a major barrier to PLC integration capabilities. The present work may be seen as a contribution to semantic skills of product data models, looking for smooth integration between Design and Manufacture stages of product's lifecycle.

From the results of this work, new research questions arise, which serve as starting points for future scientific work:

Research Direction 1 – To refine the connection and support of Ontologies with CAD files, namely through the standard STEP technology, exploring developments like those achieved in *Semantic STEP* initiatives. The implicit data semantics from STEP models could be made explicit and captured in a set of entity definitions, so that they would be able to support the information needs from other downstream applications in the product lifecycle.

Research Direction 2 – To explore the potentialities of Description Logics with an intermediary tool to express implicit manufacture specifications (and constraints), hence enriching knowledge database with particular rules of each organization – in an industrial environment, this may promote the aggregation of main process control variables or constraints and its consistent handling. For example, typical general rules existent in industrial scenario like «we do not use radius concordance bellow x micrometers» could be expressed mathematically in a simple inequality but in DL the formalism is much more robust and suitable to interact with knowledge databases.

Research Direction 3 – To define a unit of measure of the semantic richness that a Product Model may enclose; the same principle – using a *unit of though* – may be explored in the knowledge quantification of a Product Model, and to establish a metric of attained re-use.

Research Direction 4 – To explore the extension of the methodology proposed considering effective Computer-Aided Inspection Planning support. In a product Manufacture, it is very important that workpiece geometry and its deviations meet the manufacture specifications established by the precedent Phases. The inspection process analyzes and compares the data from coordinate measuring machines and describes the deviation of workpiece geometry in the coordinate systems where tolerances are specified. This reflects the quality of machining operations and it is the feedback substrate to an adequate manufacturability evaluation in each Company. Semantic properties and ontological structures may contribute to a more effective feedback inside PLC Manufacture phase.

VII. References

[1] *Exploring the concept of multiple product development via an action research project.* **Drejer, A., Gudmundsson, Agnar.** Number 3, 2003, Integrated Manufacturing Systems, Vol. 14, pp. 208-220.

[2] **Beitz, W. and Pahl, G.** *Engineering Design – A Systematic Approach.* London : Springer-Verlag, 1996.

[3] **Pugh, Stuart.** *Total design: integrated methods for successful product engineering.* Wokingham : Adddison-Wesley, 1997.

[4] **Ulrich, Karl T. and Eppinger, S. D.** *Product Design and Development.* New York : McGraw-Hill, 2000.

[5] **Kreith, Frank.** *Mechanical Engineering Handbook.* Boca Raton : CRC Press, 1999.

[6] **Systems, IMS Inteligent Manufacturing.** SMART-fm project, . *funstep.org.* [Online] 2006. www.smart-fm.funstep.org.

[7] **K., Amann.** *Product lifecycle management: empowering the future of business.* s.l. : CIM Data, Inc., 2002. p. 12.

[8] **Jardim-Gonçalves, Ricardo and Bonfatti, Flavio.** *The road for SEEM: a reference framework towards a single european electronic market.* London : CRC Press, 2008. p. 129. ISBN 978-0-415-41995-6.

[9] *Standard support for the virtual enterprise.* **Clements, P.** [ed.] NIF 227 459 822. Italy : s.n., 1997. Int. Conf. on Enterprise Integration Modeling Technology – ICEIMT'97.

[10] **Hollins, Bill and Pugh, Stuart.** *Successful Product Design.* London : Butterworth, 1990.

[11] **Monds, Fabian.** *The business of electronic product development.* London : Peter Peregrinus, Ltd, 1984. Vol. 2. ISBN 0-86341-012-X.

[12] *A hybrid systematic and conventional approach for the design and development of a product: a case study.* **Sivaloganathan, S, Cross, M Shanin, T.** 1, Great Britain : Elseviewer, January 2000, Design Studies, Vol. 21.

[13] **Back, Nelson, et al.** *Projecto Integrado de Produtos: planejamento, concepção e modelagem.* Barueri, São Paulo, : Editora Manole, Lda, 2008. p. 601. ISBN 978-85-204-2208-3.

[14] **Jones, John Chris.** *Design Methods.* 2nd ed. New York : Van Nostrand ReinHold, 1992.

[15] **Vaughan, Chris and Ishikawa, Yoshiaki.** *SC4 Industrial Data Framework.* www.tc184-sc4.org : ISO TC184/SC4/WG10/N326, 2000.

[16] *ON PROVIDING FORMAL MANUFACTURABILITY INFORMATION TO UPSTREAM PRODUCT DEVELOPMENT PROCESSES: METHODOLOGY AND CASE STUDY.* **Opiyo, Eliab Z.** Long Beach, California USA : s.n., 2005.

[17] *Concurrent Engineering: Part 1: Business requirements, life cycle activity, key agenda for success and challenge with ISO standards.* **Ishikawa, Y.** [ed.] J. Cha and Ricardo Goncalves. Funchal, Madeira : Swets & Zeitlinger, Lisse, 2003. CE 2003. Vol. 1. ISBN 90 5809 622 X.

[18] *Dados, comunicação e informação técnica na fabricação mecânica.* **Silva, João P.M.A., Pais, M. Santos and Monteiro, António A.C.** Santander, Spain : s.n., 2002, June. Proceedings of XIV Congreso Internacional de Ingeniería Gráfica INGEGRAF.

[19] **Jardim-Goncalves, Ricardo, et al.** Framework for Enhanced Interoperability Through Ontological Harmonization of Enterprise Product Models. *Ontologies: A Handbook of Principles, Concepts and Applications.* 2007.

[20] **Stark, Jonh.** *Product Lifecycle Management: 21th Century Paradigm for Product Realisation.* [ed.] Rajkumar Roy. London : Springer, 2005. p. 438. 9-78-1-85233-810-5.

[21] *A product information modeling framework for product lifecycle management.* **Sudarsan, R., et al.** 37, s.l. : Elsevier, 2005, Computer-Aided Design, pp. 1399–1411.

[22] **O'Marah, K. and Myer, B.** *The product lifecycle management applications report 2001-2006.* s.l. : AMR Research, 2002.

[23] *The application of case based reasoning to decision support in new product development.* **Belecheanu, R., et al.** 1, 2003, Integrated Manufacturing Systems, Vol. 14, pp. 36-45.

[24] **ISO.** *ISO 13584 Industrial automation systems and integration – Parts library.* s.l. : International Standard Organization.

[25] **Pithon, António.** *Projecto organizacional para a Engenharia Concorrente no âmbito das Empresas Virtuais.* Guimarães : Universidade do Minho, 2004.

[26] **Putnik, Goran.** *BM_Virtual Enterprise Arquitecture Reference Model.* [ed.] Universidade do Minho. Guimarães : Technical Report, 2000. Vols. RT-CESP-GIS-2000.

[27] **Veryhard, Richard.** *Information modelling: practical guidance.* Hertfordshire : Prentice Hall Int'l, 1992. p. 291. livro 263419 da BPG cota CDU 658.0. ISBN 0 13 454182 0.

[28] *Product modelling for conceptual design support.* **Tay, Francis and Gu, Jinxiang.** 48, 2002, Computers in Industry, pp. 143-155.

[29] *The wise approach to electronic commerce.* **Lazcano, S. and al, et.** 5, 2000, International Journal of Computer Systems Science & Engineering, Vol. 15.

[30] **West, M.** *Integration of industrial data for exchange access and sharing: A proposed architecture.* ISO. www.tc184-sc4.org : ISO TC184/SC4/WG10/N195, 1998.

[31] *Analyzing and optimizing constraint structures in complex parametric CAD models.* **Anderl, R. and Mendgen, R.** [ed.] B. Brüderlin and D. Roller. Berlin , Heidelberg :

springer-verlag, 1998, Geometric Constraint Solving and Applications.

[32] **ISO.** ISO TC184 Section 4. *ISO10303.* [Online] 2003. www.tc184-sc4.org.

[33] *Ontology-based framework for enhanced interoperability in networked industrial environments.* **Gonçalves, Ricardo Jardim, et al.** Salvador, Brazil : s.n., 2004. 11th IFAC Symposium on Information Control Problems in manufacturing.

[34] *TRENDS IN AGILE AND CO-OPERATIVE MANUFACTURING.* **Leitão, Paulo, et al.** 2001.

[35] **NIST.** [Online] http://www.mel.nist.gov/workshop/iceimt97/pap-kk1/pap-kk1.htm .

[36] *Automation, Production Systems and CIM.* **Groover, M.** s.l. : Prentice-Hall, 1987.

[37] *Agent-based Holonic Production Control.* **Leitão, Paulo and Restivo, Francisco.** Aix-en-Provence : IEEE Computer Society, 2002. Proceedings of the 3rd International Workshop on Industrial Applications of Holonic and Multi-Agent Systems. pp. pp 583-593.

[38] *Infrastructures for Virtual Enterprises:a Summary of Achievements.* **Camarinha-Matos, Luís and Afsarmanesh, H.** [ed.] L. and Afsarmanesh, H. Camarinha-Matos. s.l. : Kluwer Academic Publishers, 1999. Proceedings of the PRO-VE'99 - IFIP International Conference On Infrastructures for Virtual Enterprises. pp. pp 483–490.

[39] **Alting, Leo.** *Manufacturing Engineering Processes.* 2nd. Lyngby : Mercel Dekker, Inc, 1994. p. 491. livro BPG 200804 SDUM. ISBN 0 8247 9129 0.

[40] **Parunak, H. Van Dyke.** *What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC.* Industrial Technology Institute. s.l. : CIA'98, 1998.

[41] **Leitão, Paulo.** *An Agile and Adaptive Holonic Architecture for Manufacturing Control.* Porto : PhD thesis, 2004. PhD thesis.

[42] *Integrated and Distributed Manufacturing, a Multi-agent Perspective.* **Barata, Jorge, et al.** Enschede : s.n., 2001. Proceedings of the 3rd Workshop on European Scientific and Industrial Collaboration. pp. pp 145-156.

[43] *A Layered Approach to Distributed Manufacturing.* **Leitão, Paulo and Restivo, F.** Leuven, Belgium : Proceedings of ASI'99 International Conference, 1999.

[44] **Barata, Jorge and Camarinha-Matos, Luís.** Coalitions of Manufacturing Components for Shop Floor Agility - The CoBaSA Architecture. *International Journal of Networking and Virtual Organisations.* 2003, Vol. 2, 1.

[45] **Rabelo, R. and Camarinha-Matos, Luís.** Multi-Agent based Dynamic Scheduling. *Int. Journal on Robotics and Computer Integrated Manufacturing.* December 1994, Vol. II, 4, pp. pp 303-310.

[46] **Rabelo, F, Camarinha-Matos, Luís and Afsarmanesh, H.** Multi-Agent-Based agile Scheduling. *Journal of Robotics and Autonomous Systems.* 1999, Vol. 27, 1-2, pp. 15-28.

[47] **Modelling Language, Unified.** *www.uml.org.* 2009.

[48] **W3C.** Web Services Architecture Requirements. [Online] 2003. www.w3.org/2002/ws/.

[49] *MENTOR – A Methodology for Enterprise Reference Ontology Development.* **Sarraipa, João, Silva, João P.M.A. and Jardim-Goncalves, Ricardo.** Varna, Bulgaria : IEEE IS'08, 2008.

[50] **AAAI.** American Association for Artificial Inteligence. [Online] www.aaai.org.

[51] **Gómez-Pérez, A., Fernández-López, M. and Corcho, O.** Ontological Engineering: Methodologies and Tools. *The Second European Summer School on Ontological Engineering and the Semantic Web (SSSW-2004).* 2004.

[52] —. Technical Roadmap D.1.1.2. *OntoWeb Consortium.* 2002.

[53] *SemWeb: Proposal for an open, multifunctional multilingual system for integrated access to knowledge base about concepts and terminology.* **Soergel, Dagobert.** Washington, D.C : Advances in Knowledge Organization, Indeks Verlag, 1996. Proceedings of the Fourth International ISKO Conference 15-18 July. Vol. 5, pp. pp 165-173.

[54] *METHONTOLOGY: From Ontological Art Towards Ontological Engineering.* **Fernandez, M., Goméz-Peréz, A. and Juristo, N.** Stanford, California : s.n., 1997. Symposium on Ontological Engineering of AAAI. pp. 33-40.

[55] **Lassila, O and McGuiness, D.** *The Role of Frame based Representation on the Semantic Web.* Knowledge Systems Laboratory. Stanford, California : Stanford University, 2001. Technical Report KSL-01-02.

[56] **Patel-Schneider, PF, Hayes, P and Horrocks, Ian.** OWL Web Ontology Language Semantics and Abstract Syntax. [Online] 2003. http://www.w3.org/TR/owl-semantics/.

[57] *Federated Product Models for Enabling Simulation-Based Product Lifecycle Management.* **Bajaj, M, et al.** Orlando : ASME, 2005. Internatinal Mechanical Engineering Congress and Exposition.

[58] *Some Issues on Ontology Integration.* **Pinto, H. S., Gómez-Pérez, A. and Martins, J. P.** 1999. Proceedings of IJCAI99's Workshop on Ontologies and Problem Solving Methods: Lessons Learned and Future Trends. pp. 7.1–7.12.

[59] **Missikoff, M., et al.** D8.1 - State of the art and state of the practice including initial possible research orientations,. *Interoperability Research for Networked Enterprises Applications and Software Network of Excellence.* 2004.

[60] **Lenat, D. B. and Guha, R. V.** *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project.* s.l. : Addison Wesley, 1990.

[61] *Towards a methodology for building ontologies.* **Uschold, M and King, M.** Montreal, Canada : s.n., 1995. Proceedings IJCAI-95 Workshop on Basic Ontological Issues in Knowledge Sharing.

[62] *"On Ontologies and Enterprise Modelling".* **Fox, M. S. and Gruninger, M.** 1997. International Conference on Enterprise Integration Modelling Technology.

[63] *Building and Reusing Ontologies for Electrical Network Applications.* **Bernaras, A., Laresgoiti, I. and Corera, J.** 1996. Proceedings European Conference on Artificial Intelligence (ECAI'96).

[64] *Toward Distributed Use of Large-Scale Ontologies.* **Swartout, R., et al.** Stanford, California : s.n., 1997. Symposium on Ontological Engineering of AAAI.

[65] *Corporative memory through cooperative creation of knowledge bases and hyper-documents.* **Euzenat, J.** Banff (Canada) : s.n., 1996. Proceedings 10th KAW.

[66] *Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information.* **Decker, S., et al.** s.l. : Kluwer Academic Publisher, 1999. Semantic Issues in Multimedia Systems", Proceedings of DS-8. pp. 351-369.

[67] *FCA-Merge: Bottom-up merging of ontologies.* **Stumme, G. and Madche, A.** Seattle, WA, : s.n., 2001. 7th Intl. Conf. on Artificial Intelligence (IJCAI '01). pp. 225--230.

[68] *The PROMPT Suite: Interactive Tools for Ontology Merging and Mapping.* **Noy, N. F. and Musen, M. A.** 6, December 2003, International Journal of Human-Computer Studies, Vol. 59, pp. 983-1024.

[69] *An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies.* **Gangemi, A., Pisanelli, D. M. and Steve, G.** 1999, Data & Knowledge Engineering, Vol. 31, pp. 183-220.

[70] **De Nicola, Antonio and al, et.** Denmark : Springer, 2005. Proc. of 16th International Conference on Database and Expert Systems Applications (DEXA). pp. 655-664. LNCS no. 3588.

[71] *A survey on Ontology Mapping.* **Choi, N., Song, I. and Han, H.** 3, s.l. : SIGMOD Record, September 2006, Vol. 35.

[72] *Survey of the Observatory of European SMEs.* **Organization, The Gallup.** 196, s.l. : Flash Eurobarometer, 2006.

[73] **Camarinha-Matos, L. M. and Afsarmanesh, H.** Concept of Collaboration. [book auth.] M. M. Cunha G. Putnik. *Encyclopedia of Networked and Virtual Organizations.* s.l. : IGI, Jan 2008.

[74] **Jardim-Goncalves, Ricardo, Grilo, António and Steiger-Garcao, adolfo.** Developing interoperability in mass customization information systems. *Mass Customization Information Systems in Business.* s.l. : Idea, 2007.

[75] *Model-Driven Architecture: Vision, Standards And Emerging Technologies.* **Poole, John D.** Hungary : ECOOP 2001, 2001.

[76] *Design and Implementation of a Virtual Information System for Agile Manufacturing.* **Nagi, L.** 10, 1997, IIE Transactions on Design and Manufacturing, Vol. 29, pp. 839-857.

[77] *Implicit multi-level modeling for integration and interoperability in flexible business environments.* **Gonçalves, Ricardo Jardim and Garção, Adolfo Steiger.** Special Issue on Enterprise Components, s.l. : Communications of ACM, 2002.

[78] *Multilingual Agents: Ontologies, Languages and Abstractions.* **Willmott, Steven and et, al.** Canada : s.n., 2001. 5th Int. Conference on Autonomous Agents.

[79] *Distributed information management in virtual engineering enterprises.* **Radeke, E.** s.l. : GEN. RIDE-VE'99 IEEE Computer Society, 1999, pp. 36–43.

[80] **Davulcu, H.** Modeling and analysis of interactions in virtual enterprises. *IT for Virtual Enterprises.* 1999, pp. 12–18.

[81] *Using Meta-Model Technologies to Organize Functionalities for Active System*

Schemes. **Breton, Erwan and Bézivin, Jean.** Canada : s.n., 2001. 5th International Conference on Autonomous Agents.

[82] *A Helix-spindle model for ontological engineering.* **Kishore, R., Zhang, H. and Ramesh, R.** 2, February 2004, Vol. 47, pp. 69-75.

[83] **Noy, Natalya and McGuinness, Deborah.** *Ontology Development 101: A Guide to Creating Your First Ontology.* Stanford Knowledge Systems Laboratory. s.l. : Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, 2002.

[84] *Anchor-PROMPT: Using Non-Local Context for Semantic Matching.* **Noy, Natalia and Musen, M. A.** Seattle, USA : s.n., 2001. Workshop on Ontologies and Information Sharing at the 17th International Joint Conference in Artificial Intelligence (IJCAI-2001).

[85] *An Environment for Merging and Testing Large Ontologies.* **McGuinness, D. L., et al.** [ed.] A, Giunchiglia, F, and Selman B (eds) Cohn. 2000. KR2000: Principles of knowledge representation and reasoning. pp. 483-493.

[86] *Graph-Oriented Model for Articulation of Ontology Interdependencies.* **Mitra, P., Kersten, M. and Wiederhold, G.** s.l. : Spriger-Verlag, 2000. Proceedings of the 7th International Conference on Extending Database Technology.

[87] *Ontology Versioning and Change Detection on the Web.* **Klein, M, Kiryakov, W, Ognyanov, D, Fensel, D.** Siguenza, Spain : s.n., 2002. 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02).

[88] *Methodology for Ontology Integration.* **Pinto, H S, Martins, J P A.** s.l. : ACM Press, 2001. Proceedings of the First International conference on knowledge Capture (K-CAP 2001).

[89] *Ontology Merging for Federated Ontologies on the Semantic Web.* **Stumme, G, Maedche, A.** 2001. IJCAI-01 Workshop on Ontologies and Information Sharing. pp. 91-99.

[90] *A translation approach to portable Ontologies.* **Gruber, Thomas.** 2, s.l. : Elsevier, 1993, Knowledge Aquisition, Vol. 5.

[91] *Learning to map between ontologies on the semantic web.* **Doan, A, et.al.** Hawaii : Eleventh International WWW Conference, 2002.

[92] *OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies.* **Mena, E, et.al.** 2, s.l. : Distributed and parallel databases – An international journal, 2000, Vol. 8.

[93] *OntoManager, A workbench Environmet to facilitate Ontology Management and Interoperability.* **Hammed, A, Sleeman, D H, Preece, A.** s.l. : EON-2002: EKAW-2002 Workshop on Evaluation of Ontology-based tools, 2002.

[94] *Multilingual on-line dictionary: breaking the language barriers in the advent of open markets.* **Maló, Pedro, et, al.** Varna, Bulgária : Intelligent Systems, 2004.

[95] *ISO 68-1:1998 General purpose screw threads -- Basic profile -- Part 1: Metric screw threads.* **TC, ISO.** Geneve : ISO, 1998.

[96] *Standard parametric product data representation: What's the STEP ahead?* **Silva,**

João P.M.A., et al. [ed.] CE2003. Funchal : Balkema, July 2003, Advanced design, management and production systems. ISBN 90 5809 524 6.

[97] **Donini, F. M., et al.** A hybrid system with datalog and concept languages:. *Trends in Artificial Intelligence.* 1991, Vol. LNAI 549.

[98] *Ontologies for Enterprise Knowledge Management.* **Maedche, A., et al.** s.l. : IEEE, 2003, Journal IEEE Intelligent Systems.

[99] *Ontology mapping: the state of the art.* **Kalfoglou, Y. and Schorlemmer, M.** 1, 2003, Knowl. Eng. Rev, Vol. 18, pp. 1--31.

[100] *DL reasoner vs. first-order prover.* **Tsarkov, D. and Horrocks, I.** s.l. : CEUR (http://ceur-ws.org/). Proc. of the 2003 Description Logic Workshop (DL 2003). Vol. 81, pp. 152-159.

[101] **Arms, William.** *Lexicon definition after Digital Libraries.* s.l. : M.I.T. Press, 2000.

[102] **Brent, Michael R. and Cartwright, Timothy A.** Distribution regularity and phonotactic constraints are useful for segmentation. [book auth.] Altmann Gerry. *Psycholinguistics: Critical Concepts in Psychology.* 2002.

[103] **Navigli, R. and Velardi, P.** Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. *Computational Linguistics.* 2004, Vol. 50, 2.

[104] *Ontology-Based Framework for Enhanced Interoperability in Networked Industrial Environments.* **Jardim-Goncalves, Ricardo and Sarraipa, João.** 2004. 11th IFAC Symposium on Information Control Problems in Manufacturing - INCOM 2004.

[105] *Annotation for Enterprise Information Management Traceability.* **Sarraipa, João, et al.** Philadelphia : ASME, 2007. Proceedings of IDETC/CIE ASME 2007.

[106] **University, Stanford.** Protege ontology editor. [Online] protege.stanford.org.

[107] **Jardim-Gonçalves, Ricardo and Sarraipa, João.** *Internet-based cooperative environment for the development of the taxonomy and terminology.* s.l. : Taxonomy and Glossary for interoperability, IDEAS Roadmap, 2003. pp. pp. 104-116.

[108] *Progress and problems in extending the STEP standard.* **Pratt, M. J.** [ed.] B. Roller, D. Berlin. Berlin : Springer-Verlag, 1998, Geometric Constraint Solving and Applications. Heidelberg.

[109] *Context-explication in conceptual ontologies: the PLIB approach.* **Pierra, Guy.** Proceedings of CE2003, Funchal : Balkema, 2003.

[110] **Cross, Nigel.** *Engineering design methods : strategies for product design.* 2nd. Chichester : John Wiley, 1996. ISBN 0-471-94228-6.

[111] **computation.), (Symbolic.** *Product data interfaces in CAD-CAM applications : design, implementation and experiences.* Berlin : Springer-Verlag , 1986. ISBN 3-540-15118-4.

[112] **Niebel, Benjamin W.** *Product design and process engineering.* Tokyo : McGraw-Hill Kogakusha, 1974. ISBN 0-07-046535-5.

[113] **Jones, S. W.** *Product design and process selection.* London : Butterworths, 1973. ISBN 0-408-70413-6.

[114] *Implicit hierarchic meta-modeling - In search of adaptable interoperability for manufacturing and business systems.* **Gonçalves, Ricardo Jardim and Garção, Adolfo Steiger.** 5th IEEE/IFIP, Mexico : BASYS2002, 2002.

[115] *Modeling with constraints — theoretical foundation and applications.* **Anderl, R. and Mendgen, R.** 3, Heinemann, Oxford : Elsevier Science Limited Butterworth, 1996, Computer Aided Design, Vol. 28.

[116] *A FRAMEWORK FOR INTEGRATED COMPUTER-AIDED DESIGN, PROCESS PLANNING AND MANUFACTURING SYSTEMS ENGINEERING FOR POWERTRAIN MACHINING.* **Yip-Hoi, Derek, Li, Jianming and Zhou, Liang.** Salt Lake City : ASME, 2004. Proceedings of DETC'04 and Computers and Information in Engineering .

[117] **Brown, D.H.** [Online] Brown Associates, Inc. http://dhbrown.com.

[118] **Hammer, M., Champy, J.** *Reengineering the Corporation.* New York : Harper Collins Books, 1993.

Annex I - Bolt M6x50 STEP file sample

extract of the file produced by Autodesk Inventor 11 Professional

and after Mechanical Components libraries

Bolt M6x50 STEP Part 21 file sample

The following listing corresponds to the file recorded in the STEP Part21 format, example of instantiation of a Hexagonal Head Bolt, modeled by the Components Library of Autodesk Inventor 2009 Professional software.

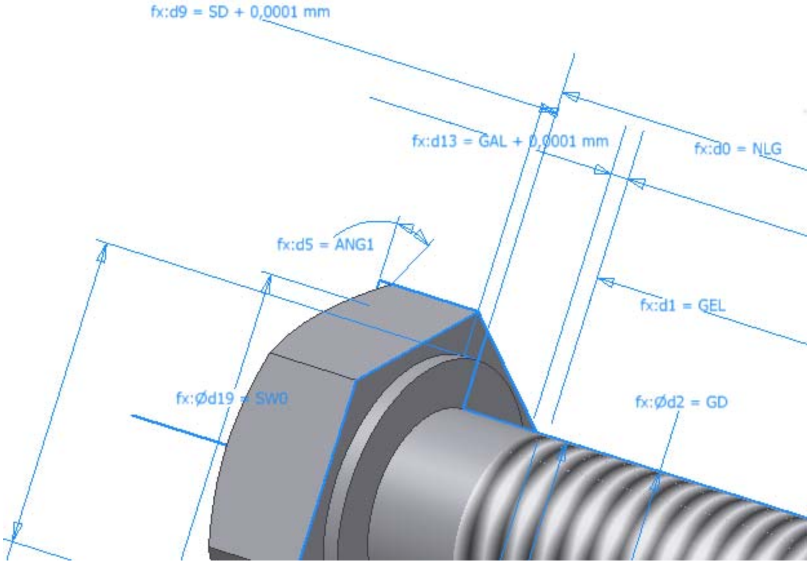


Figure AI-1 –Hexagonal Head Bolt M6x50 main variables of Autodesk library – not included in STEP Part 21 file.

```
ISO-10303-21;
HEADER;
/****************************************************************
 * Generated by software with PDE/Lib inside                     *
 * PDElib Version v51a, created Tue 12/06/2005                   *
 * International Technegroup Inc. (www.iti-oh.com)               *
 ****************************************************************/
FILE_DESCRIPTION((''),'2;1');
FILE_NAME('C:\\Inventor\\M6-50.stp','2007-12-19T19:16:10',('jpmas'),(''),'Autodesk Inventor 11','Autodesk Inventor 11','');
FILE_SCHEMA(('AUTOMOTIVE_DESIGN { 1 0 10303 214 1 1 1 1 }'));
ENDSEC;
DATA;
#5=APPLICATION_CONTEXT('automotive design');
#6=APPLICATION_PROTOCOL_DEFINITION('Draft International Standard','automotive_design',1998,#5);
#7=PRODUCT_CONTEXT('None',#5,'mechanical');
#8=PRODUCT('DIN 933  - replaced by DIN EN 24 017 M6  x 50','DIN 933  - replaced by DIN EN 24 017 M6 x 50','None',(#7));
#9=PRODUCT_RELATED_PRODUCT_CATEGORY('part','description',(#8));
#10=PRODUCT_DEFINITION_FORMATION('None','None',#8);
#11=PRODUCT_DEFINITION_CONTEXT('part definition',#5,'design');
#12=PRODUCT_DEFINITION('None','None',#10,#11);
#18=(NAMED_UNIT(*)PLANE_ANGLE_UNIT()SI_UNIT($,.RADIAN.));
#19=DIMENSIONAL_EXPONENTS(0.0,0.0,0.0,0.0,0.0,0.0,0.0);
#20=PLANE_ANGLE_MEASURE_WITH_UNIT(PLANE_ANGLE_MEASURE(0.017453292500000),#18);
```

#24=(CONVERSION_BASED_UNIT('DEGREE',#20)NAMED_UNIT(#19)PLANE_ANGLE_UNIT());
#28=(NAMED_UNIT(*)SI_UNIT($,.STERADIAN.)SOLID_ANGLE_UNIT());
#32=(LENGTH_UNIT()NAMED_UNIT(*)SI_UNIT(.MILLI.,.METRE.));
#34=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.010000000000000),#32,'DISTANCE_ACCURACY_VALUE',");
#36=(GEOMETRIC_REPRESENTATION_CONTEXT(3)GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#34))GLOBAL_UNIT_ASSIGNED_CONTEXT((#24,#28,#32))REPRESENTATION_CONTEXT('None','None'));
#37=AXIS2_PLACEMENT_3D(",#38,#39,#40);
#38=CARTESIAN_POINT(",(0.0,0.0,0.0));
#39=DIRECTION(",(0.0,0.0,1.0));
#40=DIRECTION(",(1.0,0.0,0.0));
#41=SHAPE_REPRESENTATION(",(#37),#36);
#42=PRODUCT_DEFINITION_SHAPE(",",#12);
#43=SHAPE_DEFINITION_REPRESENTATION(#42,#41);
#44=CARTESIAN_POINT(",(49.693225000000027,3.848687E-015,0.0));
#45=DIRECTION(",(-1.0,-1.209151E-016,0.0));
#46=DIRECTION(",(0.0,-1.0,0.0));
#47=AXIS2_PLACEMENT_3D(",#44,#45,#46);
#48=CONICAL_SURFACE(",#47,2.693275000000004,44.995330417327907);
#49=CARTESIAN_POINT(",(49.386450000000018,3.000000000000007,3.673819E-016));
#50=VERTEX_POINT(",#49);
#51=CARTESIAN_POINT(",(49.386450000000018,3.811593E-015,0.0));
#52=DIRECTION(",(1.0,0.0,0.0));
#53=DIRECTION(",(0.0,-1.0,0.0));
#54=AXIS2_PLACEMENT_3D(",#51,#52,#53);
#55=CIRCLE(",#54,3.000000000000003);
#56=EDGE_CURVE(",#50,#50,#55,.T.);
#57=ORIENTED_EDGE(",*,*,#56,.T.);
#58=EDGE_LOOP(",(#57));
#59=FACE_OUTER_BOUND(",#58,.T.);
#60=CARTESIAN_POINT(",(50.000000000000014,2.386550000000009,-2.922584E-016));
#61=VERTEX_POINT(",#60);
#62=CARTESIAN_POINT(",(50.000000000000014,3.885781E-015,0.0));
#63=DIRECTION(",(-1.0,0.0,0.0));
#64=DIRECTION(",(0.0,-1.0,0.0));
#65=AXIS2_PLACEMENT_3D(",#62,#63,#64);
#66=CIRCLE(",#65,2.386550000000006);
#67=EDGE_CURVE(",#61,#61,#66,.T.);
#68=ORIENTED_EDGE(",*,*,#67,.T.);
#69=EDGE_LOOP(",(#68));
#70=FACE_BOUND(",#69,.T.);
#71=ADVANCED_FACE(",(#59,#70),#48,.T.);
(…)
#618=FACE_BOUND(",#617,.T.);
#619=ADVANCED_FACE(",(#615,#618),#604,.T.);
#620=CARTESIAN_POINT(",(2.693225000000012,-1.834325E-015,0.0));
#621=DIRECTION(",(1.0,1.209151E-016,0.0));
#622=DIRECTION(",(0.0,1.0,0.0));
#623=AXIS2_PLACEMENT_3D(",#620,#621,#622);
#624=CONICAL_SURFACE(",#623,2.864683499999999,23.801970551679624);
#625=ORIENTED_EDGE(",*,*,#84,.F.);
#626=EDGE_LOOP(",(#625));
#627=FACE_OUTER_BOUND(",#626,.T.);
#628=ORIENTED_EDGE(",*,*,#612,.T.);
#629=EDGE_LOOP(",(#628));
#630=FACE_BOUND(",#629,.T.);
#631=ADVANCED_FACE(",(#627,#630),#624,.T.);

```
#632=CLOSED_SHELL('',(#71,#91,#100,#147,#192,#237,#282,#327,#370,#402,#427,#452,#496,#514,#532
,#545,#559,#579,#599,#619,#631));
#633=MANIFOLD_SOLID_BREP('',#632);
#634=COLOUR_RGB('Metal-Steel',0.639216005802155,0.639216005802155,0.686275005340576);
#635=FILL_AREA_STYLE_COLOUR('Metal-Steel',#634);
#636=FILL_AREA_STYLE('Metal-Steel',(#635));
#637=SURFACE_STYLE_FILL_AREA(#636);
#638=SURFACE_SIDE_STYLE('Metal-Steel',(#637));
#639=SURFACE_STYLE_USAGE(.BOTH.,#638);
#640=PRESENTATION_STYLE_ASSIGNMENT((#639));
#641=STYLED_ITEM('',(#640),#633);
#642=MECHANICAL_DESIGN_GEOMETRIC_PRESENTATION_REPRESENTATION('',(#641),#36);
#643=ADVANCED_BREP_SHAPE_REPRESENTATION('ABSR',(#633),#36);
#644=SHAPE_REPRESENTATION_RELATIONSHIP('SRR','None',#643,#41);
ENDSEC;
END-ISO-10303-21;
```

Annex II - Fasteners ontology

Fasteners_reference ontology

The following lists the recorded in the file *Fasteners.pont*, the ontology made in Protegé.

```
; Mon May 18 13:01:32 BST 2009
;
;+ (version "3.3")
;+ (build "Build 408")


(defclass %3ACLIPS_TOP_LEVEL_SLOT_CLASS "Fake class to save top-level slot information"
    (is-a USER)
    (role abstract)
    (single-slot Has+Shank
        (type INSTANCE)
;+      (allowed-classes Shank)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Has+Finishing
        (type SYMBOL)
        (allowed-values None Black Mate Zinc_plated)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Engagement+lenght
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot KB_252138_Slot_38
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (single-slot Has+Thread
        (type INSTANCE)
;+      (allowed-classes Thread)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Driving+Surface
        (type SYMBOL)
        (allowed-values Interior Exterior)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Height
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Pitch
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Thread+Genre
        (type SYMBOL)
        (allowed-values M W UNL UNF)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Driving+Direction
        (type SYMBOL)
```

```
        (allowed-values Left Right)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot strenght
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Material+Type
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (multislot Has+Driving+Features
        (type INSTANCE)
;+      (allowed-classes Driving+Features)
        (create-accessor read-write))
    (single-slot Code
        (type STRING)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Lenght
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Threaded+length
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot radius
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Class
;+      (comment "defines the properties of the bolt or screw in terms of mechanical properties")
        (type STRING)
;+      (value "4.8" "8.8" "10.9")
;+      (cardinality 0 1)
        (create-accessor read-write))
    (single-slot Firm
        (type STRING)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot KB_252138_Slot_37
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (single-slot Has+Material
        (type INSTANCE)
;+      (allowed-classes Material)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Nominal+Diameter
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Has_Head
        (type INSTANCE)
;+      (allowed-classes Head)
;+      (cardinality 0 1)
```

```
        (create-accessor read-write))
    (single-slot Hexa_Angle
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Inventor_Variable
;+      (comment "variable that Inventor uses to compute values in the virtual model")
        (type STRING)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (multislot produces
        (type INSTANCE)
;+      (allowed-classes Products)
        (create-accessor read-write))
    (single-slot Related+to+Material
        (type INSTANCE)
;+      (allowed-classes Material)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (single-slot Has+Standard
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (multislot Type
        (type INSTANCE)
;+      (allowed-classes)
        (cardinality 1 ?VARIABLE)
        (create-accessor read-write)))

(defclass Features
    (is-a USER)
    (role concrete)
    (single-slot Has+Standard
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write)))

(defclass Driving+Features "[ISO 4017]"
    (is-a Features)
    (role concrete)
    (single-slot Height
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Nominal+Diameter
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass Hexagonal
    (is-a Driving+Features)
    (role concrete)
    (single-slot Hexa_Angle
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass Philips
```

```
    (is-a Driving+Features)
    (role concrete))

(defclass Umbraco
    (is-a Driving+Features)
    (role concrete))

(defclass Shank
    (is-a Features)
    (role concrete)
    (single-slot Engagement+lenght
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Lenght
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Threaded+length
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass Thread
    (is-a Features)
    (role concrete)
    (single-slot Thread+Genre
        (type SYMBOL)
        (allowed-values M W UNL UNF)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Driving+Direction
        (type SYMBOL)
        (allowed-values Left Right)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Lenght
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Driving+Surface
        (type SYMBOL)
        (allowed-values Interior Exterior)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Pitch
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Nominal+Diameter
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass Material
    (is-a Features)
    (role concrete)
    (single-slot strenght
```

```
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Material+Type
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write)))

(defclass Head
    (is-a Features)
    (role concrete)
    (single-slot Lenght
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Height
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Nominal+Diameter
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass Washer "esta propriedade é existente por definição da 1899 e outra como a 4014 e 4017 - ou seja,
a partir do momento que e hexagon bolt, tem washer"
    (is-a Features)
    (role concrete)
    (single-slot radius
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Height
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Nominal+Diameter
        (type FLOAT)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass Products
    (is-a USER)
    (role concrete))

(defclass Fasteners
    (is-a Products)
    (role concrete)
    (single-slot Has+Finishing
        (type SYMBOL)
        (allowed-values None Black Mate Zinc_plated)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Has+Standard
        (type STRING)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (single-slot Has+Material
```

```
        (type INSTANCE)
;+      (allowed-classes Material)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass Threaded
    (is-a Fasteners)
    (role concrete)
    (multislot Has+Driving+Features
        (type INSTANCE)
;+      (allowed-classes Driving+Features)
        (create-accessor read-write))
    (single-slot Has+Thread
        (type INSTANCE)
;+      (allowed-classes Thread)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass screw
    (is-a Threaded)
    (role concrete)
    (single-slot Has+Shank
        (type INSTANCE)
;+      (allowed-classes Shank)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Has_Head
        (type INSTANCE)
;+      (allowed-classes Head)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (single-slot Class
;+      (comment "defines the properties of the bolt or screw in terms of mechanical properties")
        (type STRING)
;+      (value "4.8" "8.8" "10.9")
;+      (cardinality 0 1)
        (create-accessor read-write)))

(defclass bolt
    (is-a Threaded)
    (role concrete)
    (single-slot Has+Shank
        (type INSTANCE)
;+      (allowed-classes Shank)
;+      (cardinality 1 1)
        (create-accessor read-write))
    (single-slot Has_Head
        (type INSTANCE)
;+      (allowed-classes Head)
;+      (cardinality 0 1)
        (create-accessor read-write))
    (single-slot Class
;+      (comment "defines the properties of the bolt or screw in terms of mechanical properties")
        (type STRING)
;+      (value "4.8" "8.8" "10.9")
;+      (cardinality 0 1)
        (create-accessor read-write)))
```

```
(defclass nut
    (is-a Threaded)
    (role concrete))

(defclass Non+Threaded
    (is-a Fasteners)
    (role concrete))

(defclass rivet
    (is-a Non+Threaded)
    (role concrete)
    (single-slot Has+Shank
        (type INSTANCE)
;+      (allowed-classes Shank)
;+      (cardinality 1 1)
        (create-accessor read-write)))

(defclass ring
    (is-a Non+Threaded)
    (role concrete))

(defclass Manufacturers
    (is-a USER)
    (role concrete)
    (multislot produces
        (type INSTANCE)
;+      (allowed-classes Products)
        (create-accessor read-write))
    (single-slot Firm
        (type STRING)
;+      (cardinality 1 1)
        (create-accessor read-write)))
```

Annex III - Fasteners Ontology overview

Fasteners Ontology overview

The reference ontology *Fasteners_reference* (explicitly listed in Annex II) was resultant of two other ontologies harmonization. Both initial ontologies were modeled after mechanical catalogues of fasteners, using the Protégé editor. The two were selected after catalogues or standards, and suppliers, retailers and costumers' consultation as well. Both were considered as the most significant in terms of illustration of harmonization issues and mismatches.

Herein included snapshots of Protégé tool with resultant *Fasteners_reference* ontology have the solely human readability purposes.

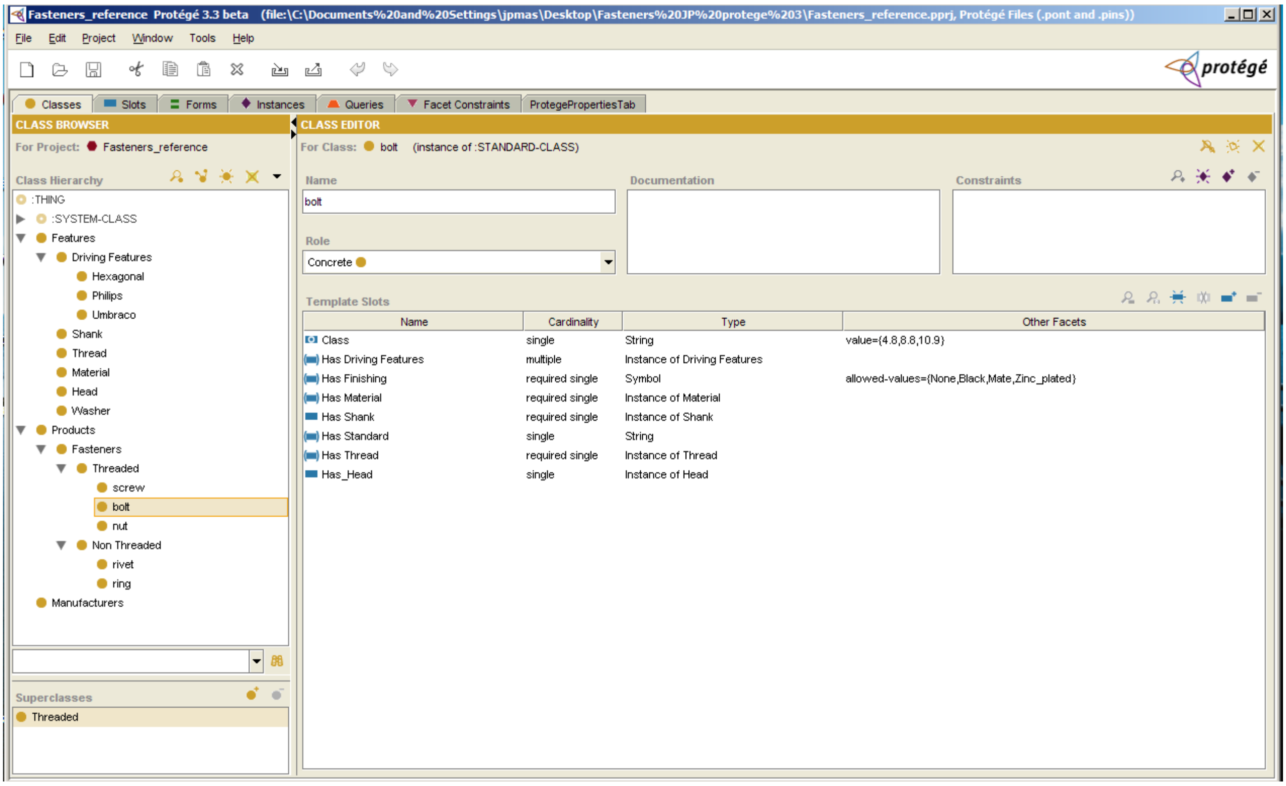


Figure AIII-1 – Overall picture of Fasteners_reference Protégé ontology editor (Classes tab, with the browser and editor pane).

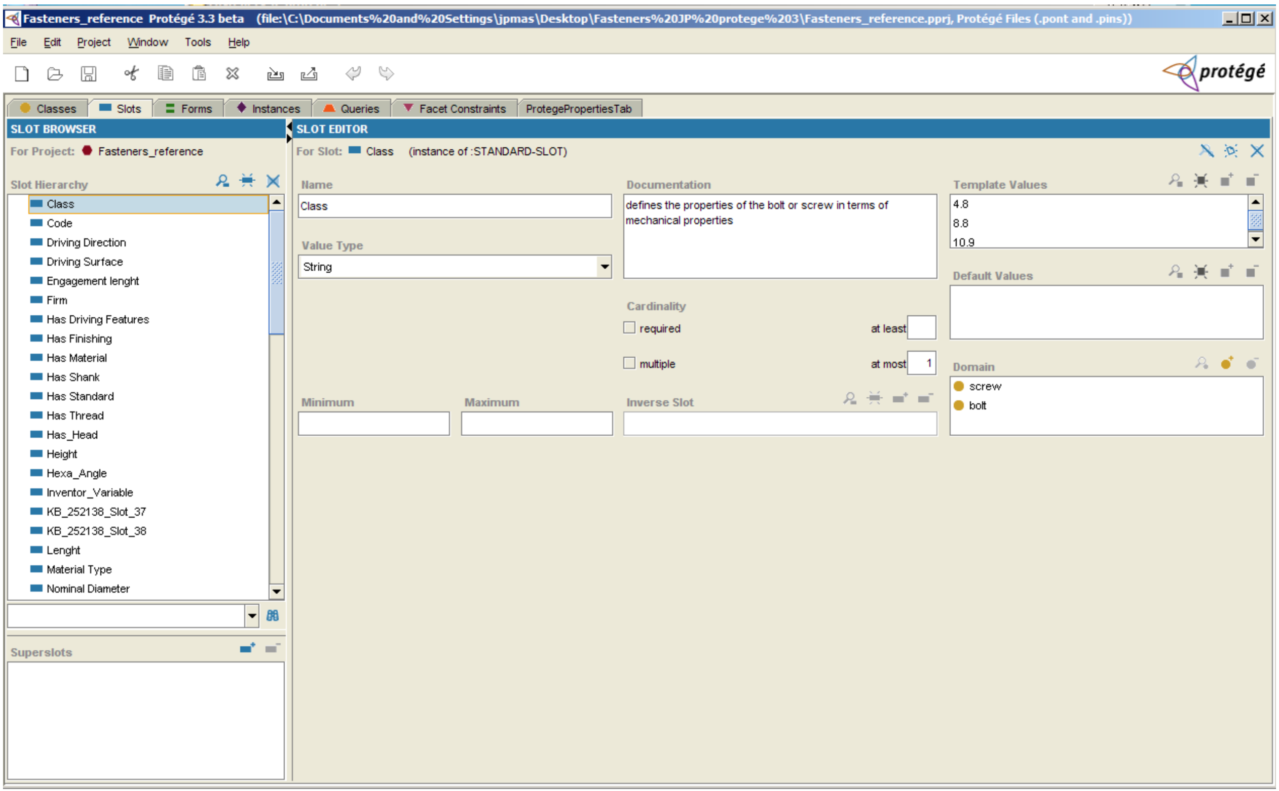


Figure AIII-2 – Properties tab of *Fasteners_reference* Protégé ontology editor.

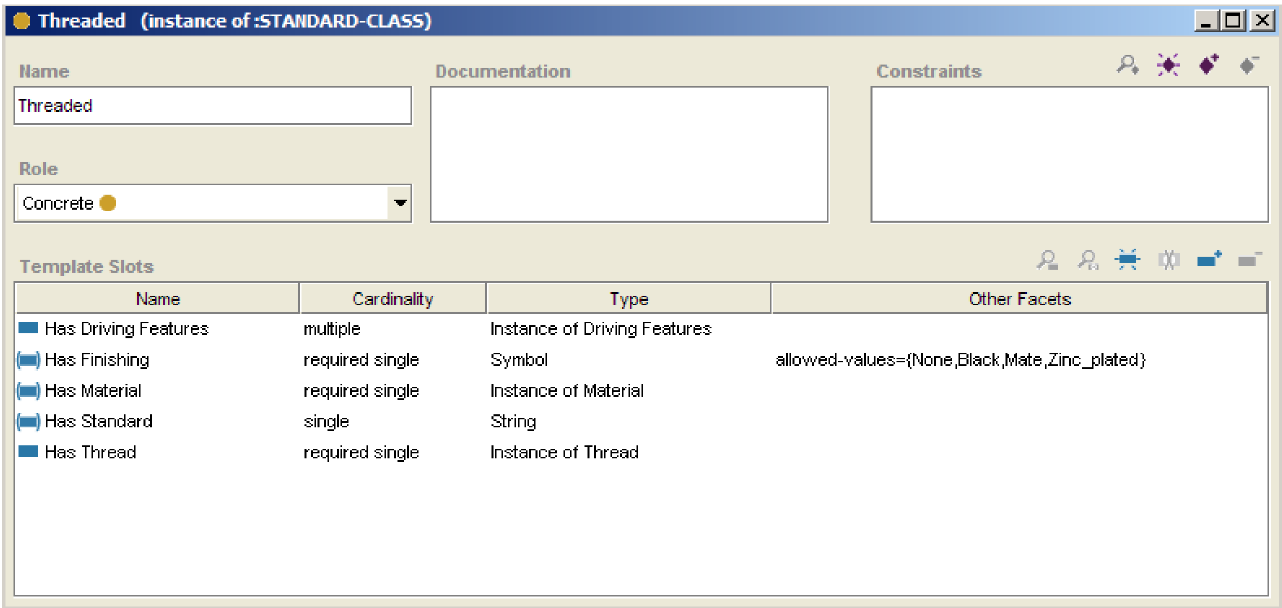


Figure AIII-3 – Inheritance of properties within Threaded Class.

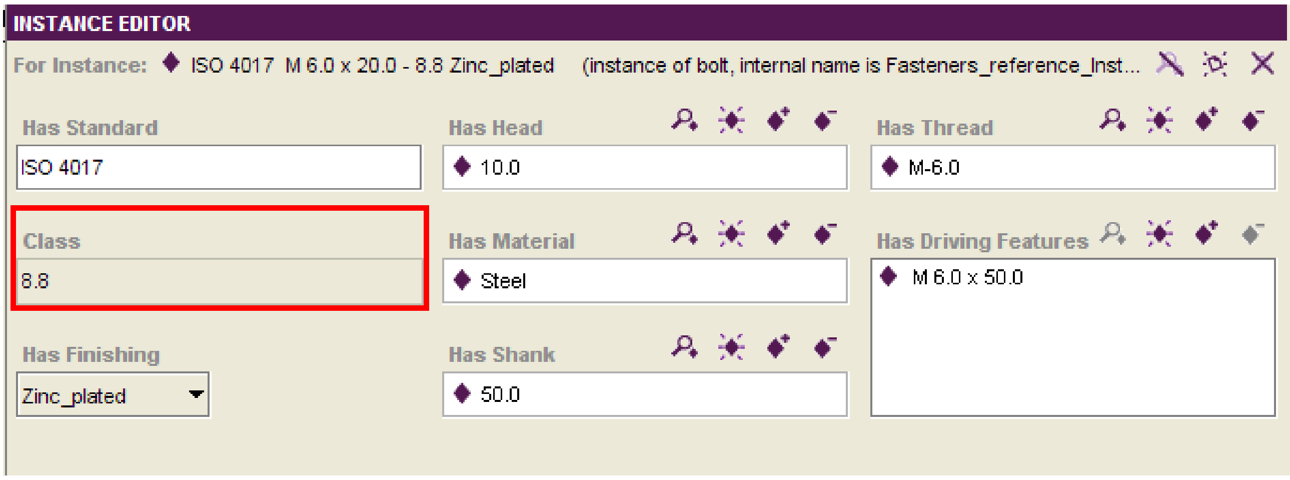


Figure AIII-4 – Instance of a hexagonal head bolt ISO 4017 M6x50.

Annex IV - Mediator Ontology excerpt

(instances not included)

Mediator Ontology excerpt

In the listing below these lines, the Mediator Ontology core could be observed. The instances that the ontology includes were suppressed on purpose, since there is no added value to human examination in this format.


```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:protege="http://protege.stanford.edu/plugins/owl/protege#"
    xmlns:xsp="http://www.owl-ontologies.com/2005/08/07/xsp.owl#"
    xmlns="http://www.owl-ontologies.com/Ontology1167819765.owl#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:swrl="http://www.w3.org/2003/11/swrl#"
    xmlns:swrlb="http://www.w3.org/2003/11/swrlb#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.owl-ontologies.com/Ontology1167819765.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Entities">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:onProperty>
          <owl:ObjectProperty rdf:ID="entity_information"/>
        </owl:onProperty>
        <owl:cardinality rdf:datatype="http://www.w3.org/2001/XMLSchema#int"
        >1</owl:cardinality>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Ontology_Characteristics"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Ontology_operations">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Operations"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Ontology_Traceability"/>
  <owl:Class rdf:ID="Information">
    <rdfs:subClassOf rdf:resource="#Ontology_Characteristics"/>
  </owl:Class>
  <owl:Class rdf:ID="Ontology_information">
    <rdfs:subClassOf rdf:resource="#Information"/>
  </owl:Class>
  <owl:Class rdf:ID="Entities_information">
    <rdfs:subClassOf rdf:resource="#Information"/>
  </owl:Class>
  <owl:Class rdf:about="#Operations">
    <rdfs:subClassOf rdf:resource="#Ontology_Characteristics"/>
  </owl:Class>
  <owl:Class rdf:ID="Entities_operations">
    <rdfs:subClassOf rdf:resource="#Operations"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="Entity_of_Ontology">
```

```
  <rdfs:range rdf:resource="#Ontology_information"/>
  <rdfs:domain rdf:resource="#Entities_information"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="ontology_operation">
  <rdfs:range rdf:resource="#Ontology_information"/>
  <rdfs:domain rdf:resource="#Ontology_Traceability"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="entity_operation">
  <rdfs:domain rdf:resource="#Entities"/>
  <rdfs:range rdf:resource="#Entities_operations"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="entity_traceability_element">
  <rdfs:domain rdf:resource="#Entities"/>
  <rdfs:range rdf:resource="#Entities"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="forward_ontology_traceability_element">
  <rdfs:range rdf:resource="#Ontology_Traceability"/>
  <rdfs:domain rdf:resource="#Ontology_Traceability"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="to_Entity">
  <rdfs:domain rdf:resource="#Entities_operations"/>
  <rdfs:range rdf:resource="#Entities"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="from_Entity">
  <rdfs:range rdf:resource="#Entities"/>
  <rdfs:domain rdf:resource="#Entities_operations"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="ontology_information">
  <rdfs:range rdf:resource="#Ontology_information"/>
  <rdfs:domain rdf:resource="#Ontology_Traceability"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="ontology_entities">
  <rdfs:domain rdf:resource="#Ontology_Traceability"/>
  <rdfs:range rdf:resource="#Entities"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="backward_ontology_traceability_element">
  <rdfs:range rdf:resource="#Ontology_Traceability"/>
  <rdfs:domain rdf:resource="#Ontology_Traceability"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#entity_information">
  <rdfs:range rdf:resource="#Entities_information"/>
  <rdfs:domain rdf:resource="#Entities"/>
</owl:ObjectProperty>
<owl:DatatypeProperty rdf:ID="Transformation_expression">
  <rdfs:domain rdf:resource="#Entities_operations"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Ontology_language">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Ontology_information"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Author">
  <rdfs:domain rdf:resource="#Operations"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Entity_type">
  <rdfs:range>
    <owl:DataRange>
```

```
    <owl:oneOf rdf:parseType="Resource">
      <rdf:rest rdf:parseType="Resource">
        <rdf:rest rdf:parseType="Resource">
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Instance</rdf:first>
          <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        </rdf:rest>
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Property</rdf:first>
      </rdf:rest>
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Class</rdf:first>
    </owl:oneOf>
  </owl:DataRange>
 </rdfs:range>
 <rdfs:domain rdf:resource="#Entities_information"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Version">
 <rdfs:domain rdf:resource="#Ontology_information"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Natural_Language">
 <rdfs:range>
  <owl:DataRange>
    <owl:oneOf rdf:parseType="Resource">
      <rdf:rest rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Portuguese</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Spanish</rdf:first>
            <rdf:rest rdf:parseType="Resource">
              <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
              >German</rdf:first>
              <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
            </rdf:rest>
          </rdf:rest>
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >French</rdf:first>
        </rdf:rest>
      </rdf:rest>
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >English</rdf:first>
    </owl:oneOf>
  </owl:DataRange>
 </rdfs:range>
 <rdfs:domain rdf:resource="#Entities_information"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Type_of_operations">
 <rdfs:domain rdf:resource="#Operations"/>
 <rdfs:range>
  <owl:DataRange>
    <owl:oneOf rdf:parseType="Resource">
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >mapping</rdf:first>
      <rdf:rest rdf:parseType="Resource">
        <rdf:rest rdf:parseType="Resource">
```

```
          <rdf:rest rdf:parseType="Resource">
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >matching</rdf:first>
            <rdf:rest rdf:parseType="Resource">
              <rdf:rest rdf:parseType="Resource">
                <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                >merging</rdf:first>
                <rdf:rest rdf:parseType="Resource">
                  <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                  >integrating</rdf:first>
                  <rdf:rest rdf:parseType="Resource">
                    <rdf:rest rdf:parseType="Resource">
                      <rdf:rest rdf:parseType="Resource">
                        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
                        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                        >versioning</rdf:first>
                      </rdf:rest>
                      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                      >evolution</rdf:first>
                    </rdf:rest>
                    <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
                    >checking</rdf:first>
                  </rdf:rest>
                </rdf:rest>
              </rdf:rest>
              <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
              >translation</rdf:first>
            </rdf:rest>
          </rdf:rest>
          <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >alignment</rdf:first>
        </rdf:rest>
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >transformation</rdf:first>
      </rdf:rest>
    </owl:oneOf>
   </owl:DataRange>
  </rdfs:range>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="URL">
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  <rdfs:domain rdf:resource="#Ontology_information"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Owner">
  <rdfs:domain rdf:resource="#Ontology_information"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Type_of_mismatches">
  <rdfs:range>
    <owl:DataRange>
      <owl:oneOf rdf:parseType="Resource">
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Disjoint</rdf:first>
        <rdf:rest rdf:parseType="Resource">
          <rdf:rest rdf:parseType="Resource">
            <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Included</rdf:first>
            <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
```

```xml
        </rdf:rest>
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Coincident</rdf:first>
      </rdf:rest>
    </owl:oneOf>
  </owl:DataRange>
 </rdfs:range>
 <rdfs:domain rdf:resource="#Entities_operations"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Value">
 <rdfs:domain rdf:resource="#Entities_information"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Domain">
 <rdfs:domain rdf:resource="#Entities_information"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
 >Presently class domain where this entity is from</rdfs:comment>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Root">
 <rdfs:range>
  <owl:DataRange>
    <owl:oneOf rdf:parseType="Resource">
      <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
      >true</rdf:first>
      <rdf:rest rdf:parseType="Resource">
        <rdf:rest rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#nil"/>
        <rdf:first rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
        >false</rdf:first>
      </rdf:rest>
    </owl:oneOf>
  </owl:DataRange>
 </rdfs:range>
 <rdfs:domain rdf:resource="#Information"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Description">
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdfs:domain>
  <owl:Class>
    <owl:unionOf rdf:parseType="Collection">
      <owl:Class rdf:about="#Ontology_information"/>
      <owl:Class rdf:about="#Operations"/>
    </owl:unionOf>
  </owl:Class>
 </rdfs:domain>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Annotation">
 <rdfs:domain rdf:resource="#Entities_information"/>
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>
<owl:DatatypeProperty rdf:ID="Name">
 <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
 <rdfs:domain rdf:resource="#Information"/>
</owl:DatatypeProperty>
<Ontology_Traceability rdf:ID="Ontology_Traceability_19">
 <backward_ontology_traceability_element>
  <Ontology_Traceability rdf:ID="Ontology_Traceability_2">
    <ontology_entities>
```

```
<Entities rdf:ID="Entities_4">
 <entity_operation>
  <Entities_operations rdf:ID="Entities_operations_6">
   <Type_of_operations rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >mapping</Type_of_operations>
   <to_Entity rdf:resource="#Entities_4"/>
   <from_Entity>
    <Entities rdf:ID="Entities_24">
     <entity_information>
      <Entities_information rdf:ID="Entities_information_25">
       <Entity_of_Ontology>
        <Ontology_information rdf:ID="Ontology_information_21">
         <URL rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
         >www.organisation.com</URL>
         <Ontology_language rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
         >OWL</Ontology_language>
         <Version rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
         >1.0</Version>
         <Owner rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
         >Organisation</Owner>
         <Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
         >REF</Name>
         <Root rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
         >false</Root>
        </Ontology_information>
       </Entity_of_Ontology>
       <Domain rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
       >Features</Domain>
       <Entity_type rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
       >Property</Entity_type>
       <Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
       >Nominal diameter</Name>
       <Root rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
       >true</Root>
       <Value rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
       >10</Value>
      </Entities_information>
     </entity_information>
    </Entities>
   </from_Entity>
   <Transformation_expression rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >1</Transformation_expression>
  </Entities_operations>
 </entity_operation>
 <entity_information>
  <Entities_information rdf:ID="Entities_information_5">
   <Domain rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >Thread</Domain>
   <Entity_type rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
   >Class</Entity_type>
   <Entity_of_Ontology>
    <Ontology_information rdf:ID="Ontology_information_1">
     <Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
     >Enterprise A</Name>
     <Root rdf:datatype="http://www.w3.org/2001/XMLSchema#boolean"
     >true</Root>
     <Ontology_language rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
     >RDF-S</Ontology_language>
```

```xml
          <Version rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
          >1.0</Version>
          <Owner rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
          >Retailer</Owner>
        </Ontology_information>
      </Entity_of_Ontology>
      <Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >nominal diameter</Name>
    </Entities_information>
   </entity_information>
   <entity_traceability_element rdf:resource="#Entities_24"/>
  </Entities>
 </ontology_entities>
 <ontology_entities>
  <Entities rdf:ID="Entities_17">
   <entity_information>
    <Entities_information rdf:ID="Entities_information_18">
      <Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >lower tolerance</Name>
      <Entity_type rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Property</Entity_type>
      <Domain rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Tolerance</Domain>
      <Entity_of_Ontology rdf:resource="#Ontology_information_1"/>
    </Entities_information>
   </entity_information>
  </Entities>
 </ontology_entities>
 <ontology_entities>
  <Entities rdf:ID="Entities_15">
   <entity_information>
    <Entities_information rdf:ID="Entities_information_16">
      <Domain rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Tolerance</Domain>
      <Entity_of_Ontology rdf:resource="#Ontology_information_1"/>
      <Entity_type rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >Property</Entity_type>
      <Name rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
      >upper tolerance</Name>
    </Entities_information>
   </entity_information>
  </Entities>
 </ontology_entities>
 <ontology_information rdf:resource="#Ontology_information_1"/>
</Ontology_Traceability>
</backward_ontology_traceability_element>

</Ontology_Traceability>
</rdf:RDF>

<!-- Created with Protege (with OWL Plugin 3.4, Build 506)  http://protege.stanford.edu -->
```