

# An Ontological Framework for Knowledge Management in Systems Engineering Processes

Olfa CHOURABI, Yann POLLET, Mohamed BEN AHMED  
*CNAM, CEDRIC- SEMPIA laboratory, PARIS  
RIADI laboratory, TUNISIA)*

## 1. Introduction

Systems Engineering (SE) processes comprise highly creative and knowledge-intensive tasks that involve extensive problem-solving and decision-making activities among interdisciplinary teams (Meinadier, 2002). SE projects involve the definition of multiple artifacts that present different formalization degrees, such as requirements specification, system architecture, and hardware/ software components. Transitions between the project phases stem from decision making processes supported both by generally available domain and design knowledge.

We argue that Knowledge about engineering processes constitutes one of the most valuable assets for SE organizations. Most often, this knowledge is only known implicitly, relying heavily on the personal experience background of system engineers. To fully exploit this intellectual capital, it must be made explicit and shared among project teams. Consistent and comprehensive knowledge management methods need to be applied to capture and integrate the individual knowledge items emerging in the course of a system engineering project.

Knowledge management (KM) is a scientific discipline that stems from management theory and concentrates on the systematic creation, leverage, sharing and reuse of knowledge resources in a company (awas el al, 2003). Knowledge management approaches are generally divided into personalization approaches that focus on human resources and communication, and codification approaches that emphasize the collection and organization of knowledge (McMahon et al. 2004).

In this work, we consider the latter approach for KM. Special focus is put on the comprehensive modeling of system engineering project knowledge. This knowledge partly resides in the product itself, while a lot of different types of knowledge are generated during the engineering processes. The background information such as why engineers came up with the final shape or geometry, what constraints were to be considered in engineering processes, and so on, can not be found either (Chan-Hong.P et al.2007). In other words, most of design rationale either disappear or exist partially in the form of engineering documents. The analysis of current engineering practices and supporting software tools reveals that they adequately support project information exchange and traceability, but lack essential capabilities for knowledge management and reuse.( C. Brandt et al., 2007)

The recent keen interest in ontological engineering has renewed interest in building systematic, consistent, reusable and interoperable knowledge models (Mizoguchi et al. 2000)( Kitamura, 2006).

Aiming at representing engineering knowledge explicitly and formally, and sharing and reusing this knowledge among multidisciplinary engineering teams, our work builds upon ontological engineering as a foundation for capturing implicit knowledge and as a basis of knowledge systematization.

In this chapter we present our vision about the main building blocks of a semantic framework for knowledge capitalization and reuse in Systems Engineering domain. The key idea behind our proposal is a flexible ontology-based schema with formally defined semantics to enable the capture and reuse of system engineering experiences.

The main contributions of this work can be summarized as follows:

- A generic ontological framework for system engineering knowledge modeling . The framework sets the fundamental concepts for a holistic System Engineering knowledge model involving explicit relationships between process, products, actors and domain concepts.
- A Knowledge capitalization model: we focus on problem resolution records during project execution. We address this problem through the use of the formal framework for capturing and sharing significant know-how, situated in projects context. we introduce the concept of Situated Explicit Engineering Knowledge (SEEK) as a formal structure for capturing problem resolution records and design rationale in SE projects.
- A Knowledge sharing model: we propose a semantic activation of potential relevant SEEK(s) in an engineering situation.

The chapter is structured as folowing: the next section discusses key background information about System Engineering processes and knowledge management issues in SE setting. Section 3 discusses roles and representative examples of ontological engineering in SE. In section 4, we detail the ontological framework for system engineering knowledge modelling. Section 5, presents a formal approach for Situated Explicit Engineering Knowledge capitalization and sharing. Section 6, illustrates our proposal in a transport system engineering process. Section 7, discusses relevant related work.

## 2. Problem statement

### 2.1 System Engineering processes

System Engineering (SE) is an interdisciplinary approach to enable the realization of successful systems. It is defined as an iterative problem solving process aiming at transforming user's requirements into a solution satisfying the constraints of: functionality, cost, time and quality (Meinadier, 2002). This process is usually comprised of the following seven tasks: State the problem, Investigate alternatives, Model the system, Integrate, Launch the system, Assess performance, and Re-evaluate. These functions can be summarized with the acronym SIMILAR: State, Investigate, Model, Integrate, Launch, Assess and Re-evaluate. (Bahill et al. 1998). This Systems Engineering Process is shown in Figure 1.

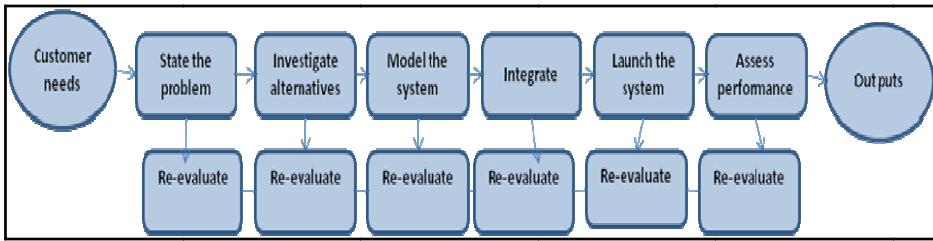


Fig. 1. The Systems Engineering Process

It is important to note that the Systems Engineering Process is not sequential. The tasks are performed in a parallel and iterative manner. At each step a comprehensive set of possible engineering models arises which are progressively combined and refined to define the target system.

Because of its inherent creative nature, it is a special case of business process. It is poorly structured and, as a rule, evolves in an unpredictable manner. In such highly dynamic settings with continuously changing requirements, the overwhelming majority of the engineering ways of working are not properly formalized, but are heavily based on the experience knowledge of the human performers.

As a consequence, engineering support environments have further to deal with the systematic collection of experience from previous project cycles and its dissemination and utilization from analogous problem solving contexts in the future. (Miatidis& Jarke, 2005). in section 4, we present a knowledge modeling framework that acts as a backend for what we expect to be a “Next generation of engineering support environment” i.e.: “knowledge centric” rather than “data centric”.

## 2.2 Knowledge Management issues in SE

The Above-delineated characteristics of SE processes show that a significant amount of knowledge is involved to solve a mix of ill- and well-defined problems. System engineers require topic knowledge (learned from text books and courses) and episodic knowledge (experience with the knowledge) (Robillard, 1991).

One of the main problems in SE processes is the lack of capture and access to knowledge underpinning the design decisions and the processes leading to those decisions (Ali-Babar et al., 2005).

System Engineers spend large portions of their time searching through vast amounts of corporate legacy data and catalogs searching for existing solutions which can be modified to solve new problems or to be assembled into a new device. This requires utilizing databases or online listings of text, images, and computer aided design (CAD) data. Browsing and navigating such collections are based on manually-constructed categorizations which are error prone, difficult to maintain, and often based on an insufficiently dense hierarchy. Search functionality is limited to inadequate keyword matching on overly simplistic attributes; it lacks the formal framework to support automated reasoning. (Miatidis& Jarke, 2005)

In this work, we focus on the knowledge modeling issue which is often considered as the first step in developing Knowledge-Based Systems (KBS). The aim of this process is to understand the types of data structures and relationships within which knowledge can be

held, and reasoned with. We use ontologies to describe the knowledge model in a formal representation language with expressive semantics.

In order to determine the basic building blocks of the knowledge repository, we introduce the notion of "SEEK" Situated Explicit Engineering Knowledge as the smallest granularity in the system experience knowledge. "SEEK", represent an integrated structure that capture product and process knowledge in engineering situations in conformance to set of layered ontologies.

### **3. Background: Ontological Engineering**

Ontologies are now in widespread use as a means formalizing domain knowledge in a way that makes it accessible, shareable and reusable (Darlington, 2008). In this section, we review relevant ontological propositions for supporting engineering processes.

In the knowledge engineering community, a definition by Gruber is widely accepted; that is, "explicit specification of conceptualization" (Gruber, 1993), where conceptualization is "a set of objects which an observer thinks exist in the world of interest and relations between them". Gruber emphasizes that ontology is used as agreement to use a shared vocabulary (ontological commitment).

The main purpose of ontology is, however, not to specify the vocabulary relating to an area of interest but to capture the underlying conceptualizations.( Gruber, 1993)

Uschold (Uschold& Gruninger, 1996) identifies the following general roles for ontologies:

- Communication between and among people and organizations.
- Inter-operability among systems.
- System Engineering Benefits: ontologies also assist in the process of building and maintaining systems, both knowledge-based and otherwise. In particular,
  - o Re-Usability: the ontology, when represented in a formal language can be (or become so by automatic translation) a re-usable and/or shared component in a software system.
  - o Reliability: a formal representation facilitates automatic consistency checking.
  - o Specification: the ontology can assist the process of identifying a specification for an IT system.

One of the deep necessities of ontologies in SE domain is, we believe, the lack of explicit description of background knowledge of modeling. There are multiple options for capturing such knowledge; we present a selection of representative efforts to capture engineering knowledge in ontologies.

(Lin et al., 1996) propose an ontology for describing products. The main decomposition is into parts, features, and parameters. Parts are defined as a component of the artifact being designed". Features are associated with parts, and can be either geometrical or functional (among others). Examples of geometrical features include holes, slots, channels, grooves, bosses, pads, etc. A functional feature describes the purpose of another feature or part. Parameters are properties of features or parts, for example: weight, color, material. Classes of parts and features are organized into an inheritance hierarchy. Instances of parts and features are connected with properties component of, feature of, and sub-feature of.

(Saaema et al., 2005) propose a method of indexing design knowledge that is based upon an empirical research study. The fundamental finding of their methodology is a comprehensive

set of root concepts required to index knowledge in design engineering domain, including four dimensions:

- The process description i.e. description of different tasks at each stage of the design process.
- The physical product to be produced, i.e. the product, components, sub-assemblies and assemblies.
- The functions that must be fulfilled by a particular component or assembly.
- The issues with regards to non functional requirement such as thrust, power, cost etc.

(Kitamura& Mizoguchi, 2004) has developed a meta-data schema for systematically representing functionality of a product based on Semantic Web technology for the management of the information content of engineering design documents.

An ontology that supports higher-level semantics is function-behaviour-structure (FBS) ontology (gero et al, 2006) . Its original focus was on representing objects specifically design artifacts. It was recently applied to represent design processes.

For ontology reusability, hierarchies are commonly established; (Borst et al, 1997) propose the PhysSys ontology as a sophisticated lattice of ontologies for engineering domain which supports multiple viewpoints on a physical system

Notwithstanding the promising results reported from existing research on SE ontologies, the reported ontological models don't provide a holistic view of the system engineering domain. They are either too generic or only focus on specific aspects of system representation.

As development of ontologies is motivated by, amongst other things, the idea of knowledge reuse and share ability, we have considered a coherent reuse of significant ontological engineering work as complementary interrelated ontologies corresponding to the multiple facets of system engineering processes.

## **4. Ontological framework for knowledge modeling in System Engineering projects**

In this section, our framework for knowledge modeling in system engineering projects is described. It structures the traces of engineering in the form of semantic descriptions based on a system engineering ontology. Section 4.1 introduces the so-called "SE general Ontology", Section 4.2. Describes the modeling layers considered for semantic knowledge capture and section 4.3 presents an engineering illustrative example.

### **4.1 SE general Ontology**

We focus here on the knowledge modeling issue that is often considered as the first step in developing a knowledge management system. The aim of this process is to understand the types of data structures and relationships within which knowledge can be held and reasoned with. We use ontologies to describe the knowledge model by a formal representation language with expressive semantics.

In order to determine the basic building blocks of the knowledge model, we introduce the notion of Situated Explicit Engineering Knowledge "SEEK" as the smallest granularity in the system experience knowledge. The systems engineering project assets represent an integrated structure that captures product and process knowledge in engineering situations

as an instance of loosely connected ontology modules that are held together by a general ontology for systems engineering.

This general ontology is developed in domain, product, and process modules. The three levels are required to provide a comprehensive semantic model for the systems engineering project asset through an integrated representation of its semantic content, its structural content, and its design rationale.

By instantiating these ontological concepts, concrete "SEEK" could be stored in a system engineering repository for future reuse. Furthermore, the ontology itself can serve as a communication base about the products and processes e.g. for exploring domain knowledge for system engineers.

**-Domain facet:** The domain ontology defines the specific domain concepts, attributes, constraints, and rules. It aims to capture formally a target system according to its different abstraction levels; in other words, for each engineering domain, the ontology defines a consensual semantic network to represent domain-specific requirements, functions, behavior and physical components, as well as their structural relationships (such as "is a" "part of") and their semantic relationships (such as "allocation"). For example, domain ontology for electric circuits might define, among other things, generic types of electric components such as transistor, connection relation among components, physical laws among physical quantities, functions of components, and allocation relations between components and functions.

Figure 3 presents a high level description of a typical domain facet.

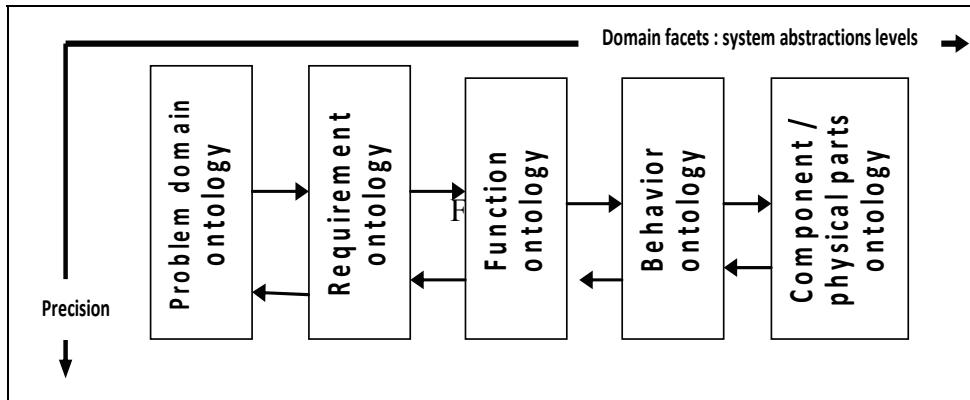


Fig. 2. Ontologies for system engineering domain facets

**-Product facet:** The product ontology contains concepts and relations that represent artifact types such as requirement documents, functional models, or conceptual schema. The product ontology provides logical structure and basic modeling constructs to describe engineering artifacts. This means that data can be extracted from domain ontology and packaged into an ontological constructed conceptual model or an engineering document. By formally relating modeling elements to domain concepts we could provide a systematic and semantic description of an engineering solution.

**-Process facet:** The process ontology contains concepts and relations that formally describe engineering activities, tasks, actors, and design rationales concepts (goals, alternatives,

arguments, and justifications for engineering decisions). Both the process and the product facets act as a formal logical structure for the systems engineering project asset. The domain facet provides semantic content for this structure.

Both the process and the product facets act as a formal structure for the SEEK. The domain facet provides semantic domain values for characterizing this structure.

Figure 4, illustrates the relationships and the complementarily of our three modeling facets for comprehensively representing system engineering knowledge.

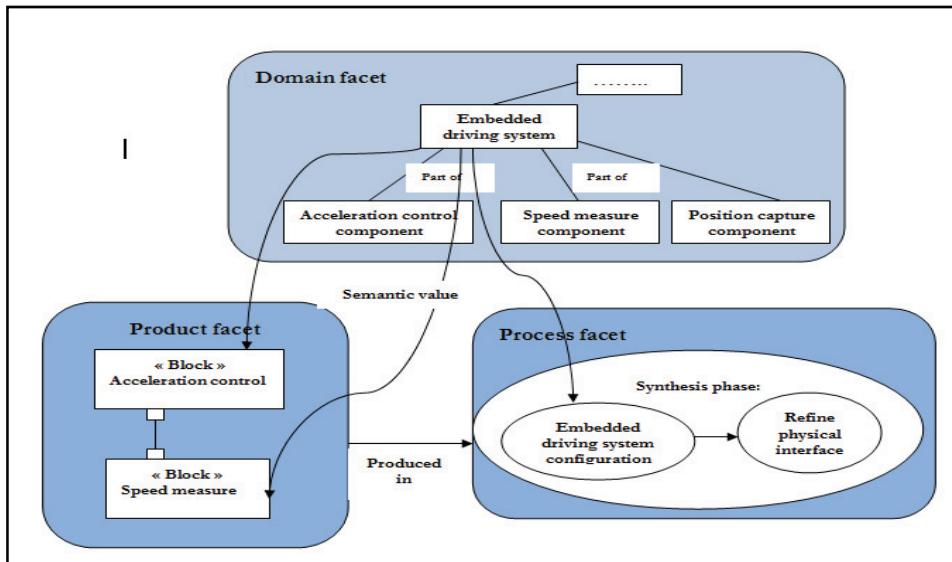


Fig. 3. Relationships between the three modeling facets in SE general ontology.

#### 4.2 Multi-layered ontologies for SE knowledge modeling

While the ontological modules for domain, product, and process introduce general-level concepts that describe a systems engineering project asset, they need to be specialized and refined in order to provide an operational knowledge model for systems engineering projects. To this end, we introduce a layered organization of these ontological modules: a general ontology for system engineering, a specialized ontology for an engineering domain (such as automotive or information systems), and an application-specific ontology. Layers subdivide the ontology into several levels of abstraction, thus separating general knowledge from knowledge about particular domains, organizations, and projects. This allows all the engineering assets to be based on generic concepts while at the same time providing a mechanism to enable different stakeholders to define their own specific terminology and concept interpretation. By instantiating the most specific ontological concepts, concrete information items can be stored in a centralized project repository. Ontological concepts act as a semantic index for engineering artifacts.

Each layer is defined along the two axes of abstraction and semantic links. Abstraction allows modeling a gradual specification of models that are more and more concrete, that is, from abstract system requirement to concrete system components. The semantic links define

how the concepts within and between an ontology module are related to each other. Typical semantic links are subsumption relations, “part of” relations and traceability relations. For example, in an ontological module for a domain, the “part of” relation could be defined on physical components assemblies and a traceability relation (allocation) could be defined to map system functions onto physical components.

Basically, knowledge in a certain layer is described in terms of the concepts in the lower layer. Figure 5 shows a hierarchy of ontologies built on top of SE general ontology.

The first layer aims to describe super-concepts that are the same across all domains, it corresponds to the SE General ontology. The domain layer defines specializing concepts and semantic relations for a system engineering domain such as aeronautics. It integrates for examples domain theories and typical domain concepts that are shared in an engineering community. The application layer, presents specialized concepts used by specific system engineering organization, this is the most specialized level for knowledge characterization and acts as a systematized representation for annotating engineering knowledge projects. The fourth layer corresponds to semantic annotation on SE project assets defined using conceptual vocabulary from the application layer. In this way, all SE project assets are captured as formal knowledge models, by instantiating these ontological concepts.

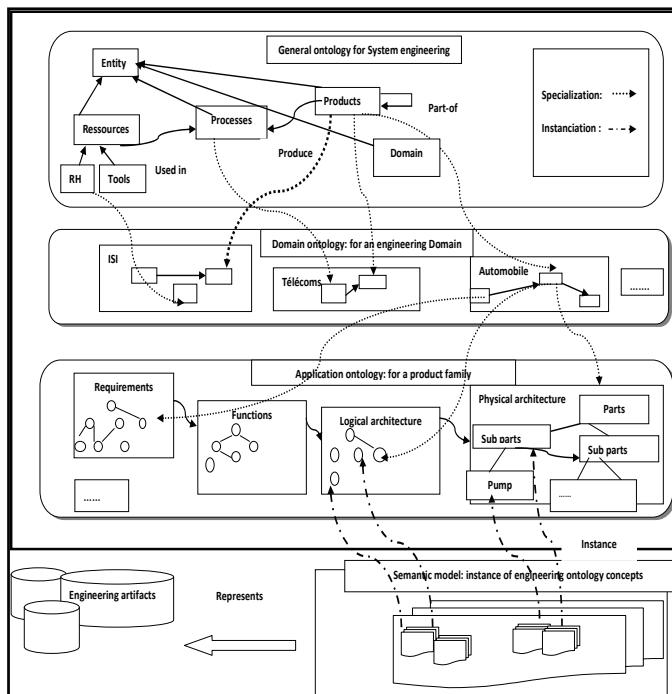


Fig. 4. Layered ontologies for systems engineering knowledge modeling

### **4.3 Illustrative example:**

We describe a knowledge modeling scenario in the domain of aeronautics engines construction. As a single scenario cannot cover all the application possibilities, we focus in this example on the formal modeling of an engineering artifact as an instance of a domain facet excerpt.

The association of a formal knowledge description to the engineering artifact in the figure 6, allows to retrieve it by a semantic search. This artifact is modeled as instances of the concepts "aircraft engine driven pump", "jet engine" and "hydraulic pump".

A query formulated with the concept "pump" allows to retrieve this engineering artifact by reasoning on the subsuming relation between "pump" and "hydraulic pump".

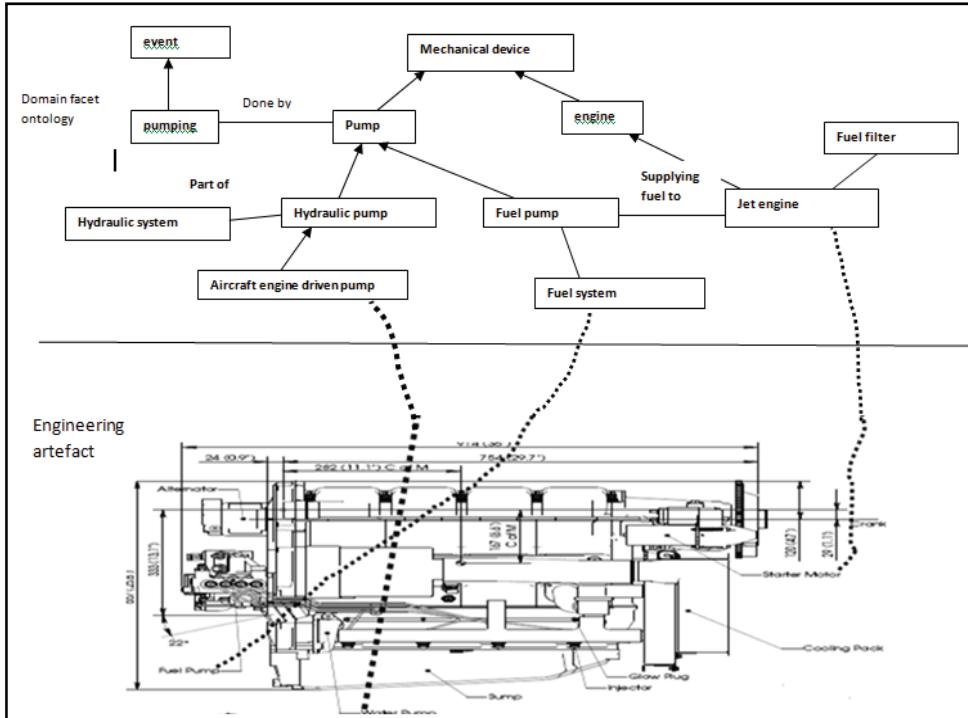


Fig 5. : Engineering artifact semantic annotation

## **5. Situated Explicit Engineering Knowledge capitalization and sharing**

In this section we further detail the experience management module of the framework. We address the dynamic aspect of engineering process with the aim to capture implicit knowledge, decision and argumentation over the proposed ontological based support for SE. Concentrating on providing knowledge items relevant to assist the human expert in solving knowledge-intensive tasks.

Our proposal relies on an explicit modeling of the relation between engineering situation, engineering goals, engineering alternatives and solutions.

The core of the model is denoted "SEEK": Situated Explicit Engineering Knowledge a formal pattern for representing knowledge in context. SEEK (s) are defined by instantiating the ontologies of the application layer.

## 5.1 Formal definitions

Before defining the Situated Explicit Engineering Knowledge, let us formalize the ontologies modules mentioned in section 4: we consider here two complementary ontologies : the system oriented ontology that corresponds to the domain facet and the context oriented ontology that corresponds to the product and the process facet.

We are representing formally ontology as:

$O = (C, \leq_C, R, \leq_R, A)$  consisting of a set of concepts  $C$  organized in a hierarchy with a subsumption relation  $\leq_C$ , a set of semantic relations organized with  $\leq_R$ , and a set of axioms stating restrictions on the conceptualization such as cardinalities, transitivity and constraints.

### Definition 1: System oriented ontology (OS)

$O^S = (C^S, \leq_C, R^S, \leq_R, A^S)$  contains the background (domain) knowledge required to engineer a system. It systematizes domain knowledge according to different system abstraction levels. System ontology is organized as a network of modular sub-ontologies representing domain-specific requirements, functions and physical components, as well as their structural relationships (such as "is a" "part of") and their semantic relationships (such as "allocation").

$O^S := (O_{\text{REQUIREMENT}}, O_{\text{FUNCTION}}, O_{\text{STRUCTURE}}, R_{\text{REQUIREMENT-FUNCTION}}, R_{\text{FUNCTION-STRUCTURE}}, R_{\text{STRUCTURE-REQUIREMENT}})$

### Definition 2: Context oriented ontology (OC)

$O^C = (C^C, \leq_C, R^C, \leq_R, A^C)$  contains knowledge required to express the circumstances under which system knowledge will be used. It consists of concepts and relations that formally describe engineering activities, roles, tools and artifacts models. Context ontology is organized as a network of modular sub-ontologies

$O^C := (O_{\text{PROCESS}}, O_{\text{TOOLS}}, O_{\text{ROLE}}, O_{\text{PRODUCTS}}, R_{\text{PROCESS-TOOLS}}, R_{\text{PROCESSUS-ROLE}}, R_{\text{PROCESSUS-PRODUCT}})$

### Definition 3: Semantic Annotation (Annot)

Annot := ( $C_a, R_a, I$ ), where:

- $C_a$  is a set of ontological concepts.
- $R_a$  is a set of ontological relations.
- $I$  is a set of tuple  $(c, r)$ , with :  $c \sqsubset C_a$  and  $r \sqsubset R_a$ .

A semantic annotation is defined as a set of ontological concepts and semantic relations instances. We use semantic annotation to express a particular modeling choice or a particular engineering situation. Figure 7, shows an example of semantic annotation defined over a system ontology fragment.

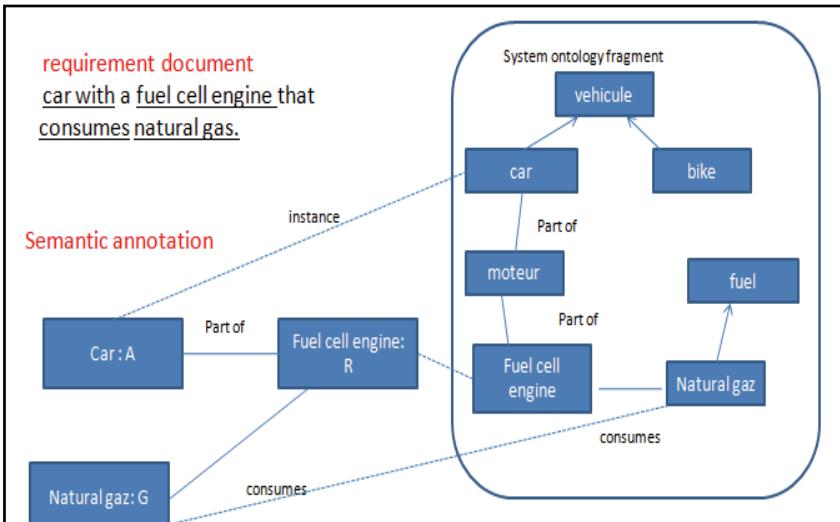


Fig. 6. Relationships between semantic annotation, ontology and engineering product.

The association of a formal knowledge description to the engineering artifact (requirement document) in the figure 7, allows to retrieve it by a semantic search. Semantic search retrieves information based on the types of the information items and the relations between them, instead of using simple String comparaisons (Brandt et al, 2007)

#### Definition 4: Situated Engineering Explicated Knowledge (SEEK)

Let OS and OC be, respectively, the system ontology and the context ontology, AnnotS and AnnotC be respectively semantic annotations over OS and OC.

A SEEK is a formal pattern for capitalizing experience knowledge.

SEEK = (EST, EG, AS, ES, REST-EG, REG-AS, RAS-ES, RES-EST) where :

- EST : Engineering Situation :=( (annotS, annotC))
- EG: engineering goal :=((annotS,annotC))
- AS: alternative solutions (annotC)
- ES: engineering solution (annotC)
- REST-EG: an engineering situation have an engineering goal
- REG-AS: an engineering goal generates alternative solutions
- RAS-ES: engineering solution choice
- Rs-p: justification against engineering situation

## 5.2 Knowledge representation model

To provide operational use of SEEK (s), must rely on solid theoretical foundations requiring an appropriate representation language, with clear and well-defined semantics.

We choose conceptual graphs (sowa, 1984) as a representation language. The attractive features of conceptual graphs have been noted previously by other knowledge engineering researchers who are using them in several applications (chein et al, 2005), (Dieng et al., 2006) (corby et al., 2006) Conceptual graphs are considered as a compromise representation

between a formal language and a graphical language because it is visual and has a sound reasoning model.

In the conceptual graph (CG) formalism, the ontological knowledge is encoded in a support. The factual knowledge is encoded in simple conceptual graphs. An extension of the original formalism (Baget, 2002) denoted “nested graphs” allows assigning to a concept node a partial internal representation in terms of simple conceptual graphs.

To represent SEEK (s) in conceptual graph formalism we use this mapping:

- The context ontology and the system ontology are represented in a conceptual graph support.
- Each semantic annotation is represented as a simple conceptual graph.
- A SEEK is a nested conceptual graph, where the concepts engineering situation, engineering goal, alternative solution, engineering solution are described by means of nested CG. This generic model has to be instantiated each time an engineering decision occurs in a project process. Figure 8 describes a SEEK model as a nested conceptual graph.

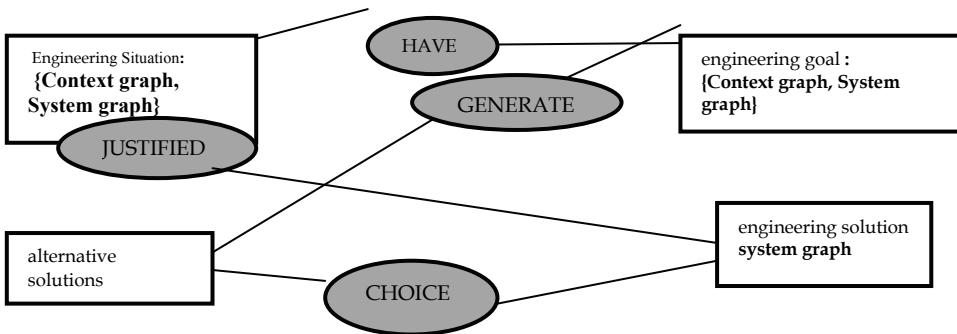


Fig 7. SEEK as a nested conceptual graph

## 5.2 Knowledge sharing model

We aim to provide a proactive support for knowledge reuse. In such approaches (Abecker et al., 1998) queries are derived from the current Work context of application tools, thus providing reusable product or process knowledge that matches the current engineering situation.

Finding a matching between an ongoing engineering situation and goal and a set of capitalized SEEK(s) relies on a standard reasoning mechanism in conceptual graphs: the projection. Let's remind the projection operation as defined by ( Mugnier & Chein, 1992)

### Mugnier and Chein Projection ( Mugnier & Chein, 1992)

Given two simple conceptual graphs G and H, a projection

from G to H is an ordered pair of mappings from  $(RG, CG)$  to  $(RH, CH)$ , such that:

- For all edges  $rc$  of G with label i,  $\Pi(r)\Pi(c)$  is an edge of H with label i.
- $\forall r \in RG, type(\Pi(r)) \leq type(r); \forall c \in CG, type(\Pi(c)) \leq type(c)$ .

There is a projection from G to H if and only if H can be derived from G by elementary specialization rules

Using the projection, the reasoning system is able to find not only descriptions of experiences that are annotated by exact concepts and relationships but also those annotated

by subtypes of these concepts. Besides, to search with imprecise and/or incomplete experiences or to answer a vague query, approximate projections can be used.

We also work on an extension to conceptual graphs projection in order to take into account partial (part-of) engineering situation matching.

Our ultimate goal consist in defining an approximate situation matching, having as result a partial ordering on the SEEK (s) according to their relevance for the current engineering situation.

## 6. Case study: automated transport sub system.

This section presents a case study of ontology based modeling for situated engineering experience. The application domain is automatic transport sub system: an automated wagon. As an example, we consider a typical component allocation process of this sub system.

We assume that the sub system's functional view is represented by the data flow diagram (DFD) depicted in figure 9. The main functions considered are: capture speed, capture position, control movement, propel, break, and contain travelers. The DFD is a result of structured functional decomposition of the initial requirement: "to transport travelers from one point to another". The design know-how including such functional knowledge used in the conceptual design phase is usually left implicit because each designer possesses it. In complex engineering domains, this implicit knowledge could play a crucial role for systematizing conceptual design. As multidisciplinary teams (mechanical, electrical, software developer..) often work concurrently on a single system, it would be beneficial to have an agreement about functional concepts describing a family of system in a unambiguous and explicit manner.

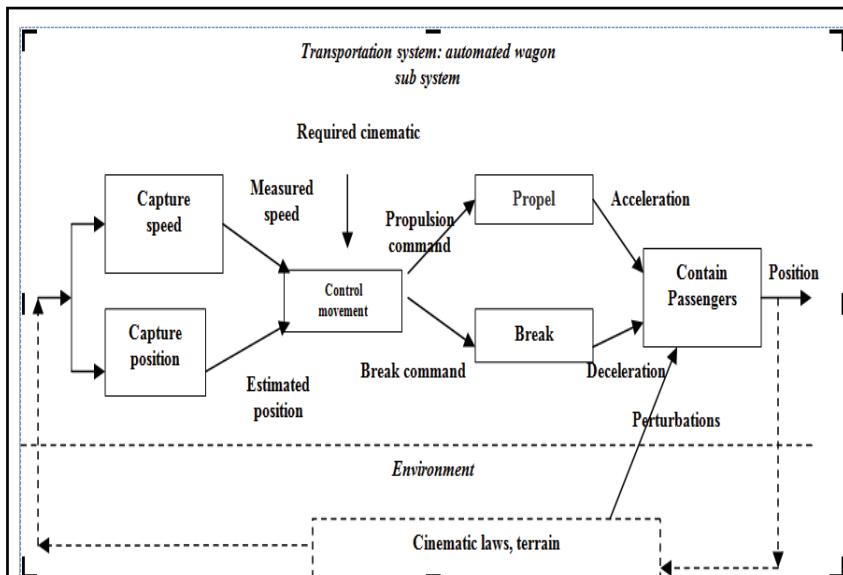


Fig. 8. Data flow diagram for an automated wagon subsystem

The system functions should be mapped to the physical components. Functions mapping to physical components can be one to one or many to one. In addition Physical solution is constrained with non functional requirements (or soft goals) such as: system performance with attributes of travel duration, facility, acceleration limitation, comfort, and reliability.

Each physical solution choice raises a set of possible engineering alternatives. The global requirements are traded-off to find the preferred alternatives. An intricate interplay usually exists among alternatives. For example, the functions speed capture and position estimation choosing inertial station that delivers the speed as well as the position, for implementing the function speed capture would restrict the engineering choices to exclude specific transducers.

Given a specific function set, it would be helpful for novices' engineers to find the corresponding physical components as well as the constraints about function allocation choices.

Like in the functions decomposition context, this knowledge is scattered in a huge mass of engineering documents, and thus not explicitly modeled. We argue that an explicit representation of the links between components and functions could improve the quality of the allocation process.

Furthermore, designers usually wish to reuse captured design knowledge to adapt past solutions and apply these to current problems, and novice designers may wish to understand lessons from previous experiences. In this case, a machine readable representation of engineering decision trace during projects should enable effective reuse of previous decisions. To address these issues, we draw upon ontological engineering for providing a systematic basis for domain engineering knowledge and using it as a foundation for describing engineering choices and tradeoffs emanating from previous projects.

To illustrate the use of the proposed ontological framework to define a Situated Engineering Knowledge, we use the ontologies excerpts depicted in figure 10 and 11 .

- The OS fragment captures the functional and the structural facet
- The OC fragment captures the process and the product facet. In this example, we adopt a recommended process (leblanc, 2008) for SE with SysML modeling language.

Using these sub ontologies we can describe the SEEK for a design decision of a speed and position capture sub system of an automated wagon, subject to a particular non functional requirement of reliability. The SEEK model is shown in figure 12.

If we consider a new engineering situation, described by the same contextual ontology, and aiming at allocating the function "move" for an automated wagon. The capitalized SEEK is matched with this engineering situation by taking the specialization relation between ontological concepts into account.

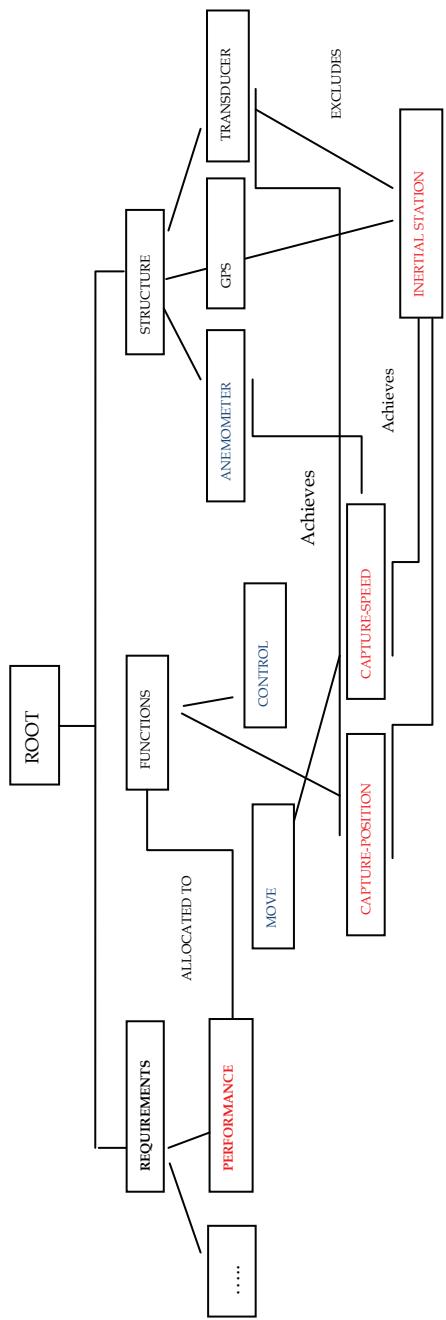


Fig. 9. a excerpt of system ontology

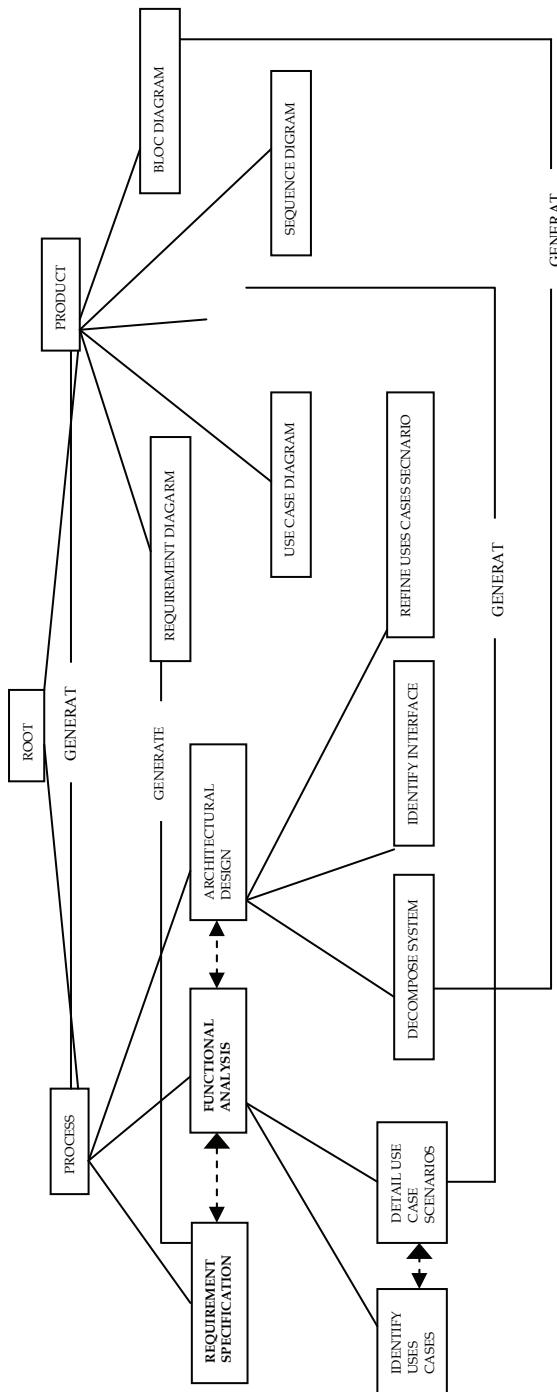


Fig. 10. an excerpt of context ontology

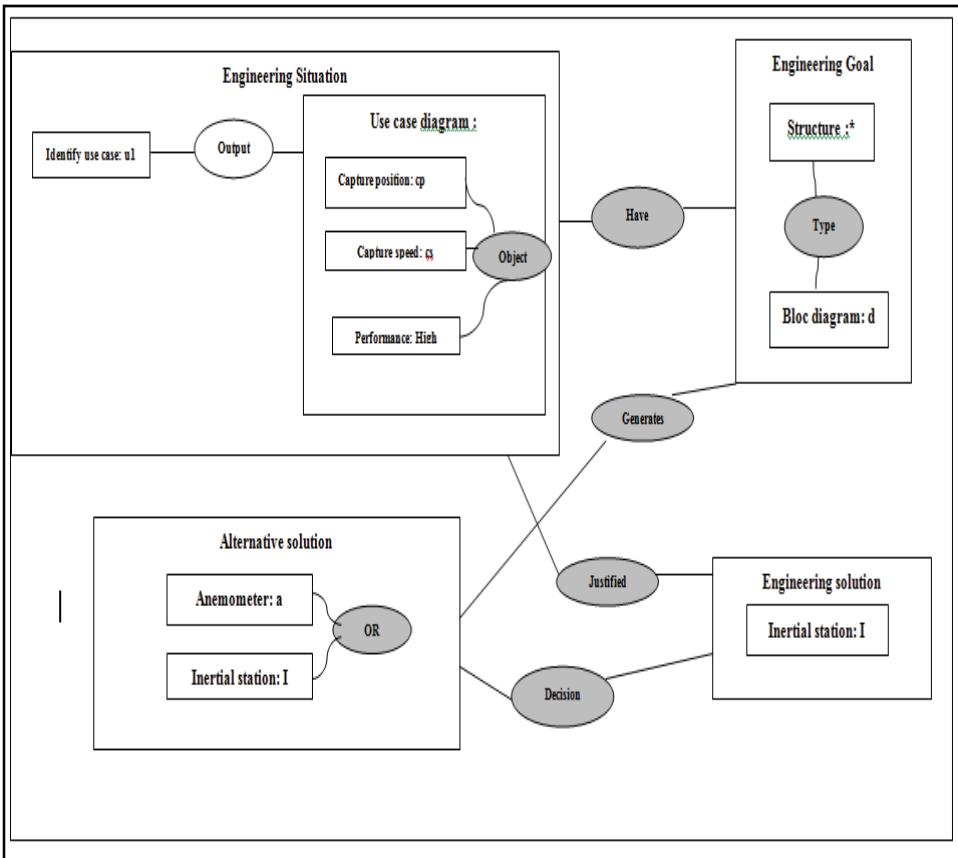


Fig. 11. Instantiated SEEK

## 7. Related work

Most of the existing SE tools still lack essential aspects needed for supporting knowledge capitalization and reuse during projects processes. Some recent research projects try to address this issue in specific engineering domains. To our knowledge, there is no generic framework for knowledge management in SE domain. We discuss, here some related works to our researches. As system engineering domain provides a generic methodological scheme to several engineering domain, we discuss some approaches from the software engineering and design engineering (sub)-domains.

In the software engineering domain Efforts such as, REMAP (Ramesh et al., 1992), REFSENO (Tautz et al., 1998) and BORE (Henninger, 1998) can be regarded as the main research stream that contributes to software knowledge management. However, the knowledge models employed by these approaches vary. REMAP and REFSENO are the closest efforts to our approach. REMAP also installs argumentations as an embedded component similar to our knowledge model, but our model extends REMAPS

characterization of what is considered to be a system engineering knowledge asset.

In the design engineering domain, (Kim et al., 2001) have integrated concepts of artificial intelligence into commercial PDM systems. The software is based on a dynamic and flexible workflow model, as opposed to the deterministic workflows seen in most commercial PDM applications. A built-in workflow management system enables the integrated management of task processes and information flows. The system can be searched by means of a semantic query and manipulation language. However, the system relies on prescriptive process definitions which require relatively well-understood and well-documented processes. As this does not apply to conceptual process engineering, this approach must be extended to ill structured processes.

(gao et al, 2003) Describes an integration of a PDM system with ontological methods and tools. The Protégé ontology editor is combined with a commercial PDM system to provide knowledge management capabilities for the conceptual design stage. In an associated research project, a graphical interface for user-friendly knowledge acquisition has been developed. In contrast to our approach experience knowledge is not recorded. Again, this approach relies on a domain where the processes are better understood than in system engineering.

(Kopena & Regli, 2003) Have designed ontology for the representation of product knowledge. A Core Ontology defines the basic structure to describe products from a functional view. Vocabulary extensions describe the domain of electromechanical devices. The model representation is based on a Description Logic (DL) system with low expressivity to achieve good computability and scalability. In this work, only product description facet is addressed.

An ontological architecture for knowledge management that resembles to our proposed framework has been proposed by ( Brandt et al. 2007) and illustrated in the context of chemical engineering processes. The application and the extension of their ontological concepts to system engineering domain were among our initial research investigation.

This survey is not exhaustive, we have discussed the important projects that guides our modeling choices, there are several recent research approaches pointing in this direction; that is exploiting ontological engineering and semantic web technologies to improve engineering processes.

## 8. Conclusion

System engineering processes implies the management of information and knowledge and could be considered as a knowledge production process. The main objective of this chapter is to present our ongoing work concerning the validation of our ontological based framework for experience capitalization and reuse. A principal strand of future research is the application of this modeling framework in the context of an engineering organization to trigger further improvement. We plan also to use the same framework for capturing “best practices” knowledge. The problem of providing a knowledge management interface integrated to existing system engineering support tools is also under investigation.

## **9. References**

- J.Meinadier (2002). Le métier d'intégration de systèmes. Hermes Science Publications.
- Awad, E. M., & Ghaziri, H. M..(2003) Knowledge management. Prentice Hall.
- McMahon, C., Lowe, A., & Culley, S. J. (2004). Knowledge management in engineering design, personalisation and codification. *Journal of Engineering. Design*, 15(4), 307-325
- Chan-Hong park, M.Kang, J.Ahn, (2007).A knowledge representation framework in design repositories. *Solid state phenomena Vol 120* , pp.235-240.
- Sebastian C. Brandt, Jan Morbach, Michalis Miatidis ,Manfred Theissen, Matthias Jarke , Wolfgang Marquardt. (2007). An ontology-based approach to knowledge management in design processes. *Computers and Chemical Engineering*
- Riichiro Mizoguchi, Yoshinobu Kitamura.(2000) "Foundation of Knowledge Systematization: Role of Ontological Engineering". The Institute of Scientific and Industrial Research, Osaka University, 8-1, Mihogaoka, Ibaraki, Osaka, 567-0047, Japan.
- Yoshinobu Kitamura. (2006) "Roles Of Ontologies Of Engineering Artifacts For Design Knowledge Modeling". In Proc. of the 5th International Seminar and Workshop Engineering Design in Integrated Product Development. (EDIPoD 2006), 21-23 September 2006, Gronów, Poland, pp. 59-69.
- T. Bahill and B. Gissing (1998).Re-evaluating systems engineering concepts using systems thinking, *IEEE Transaction on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 28 (4), 516-527.
- Michalis Miatidis, Matthias Jarke. (2005) "Toward Improvement-Oriented Reuse of Experience in Engineering Design Processes". R. Khosla et al. (Eds.): KES 2005, LNAI 3684, pp. 228-234.
- Robillard, P.N.( 1991) The role of knolwedge in software development. *Communications of the ACM*,42(1): p. 87-92.
- Ali-Babar, M., I. Gorton, and B. Kitchenham,( 2005). A Framework for Supporting Architecture Knowledge and Rationale Management, in *Rationale Management in Software Engineering*, A.H. Dutoit, et al., Editors.
- Darlington, M. & Culley, S. (2008). Investigating Ontology Development for Engineering Design Support. *Advanced Engineering Informatics*, 22(1): 112-134.
- GRUBER, T. R.( 1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5 (2): 199-220.
- M. Uschold, M. Gruninger, (1996). Ontologies: principles, methods and applications, *Knowledge Engineering Review* 11 (2) , June.
- J. Lin, M.S. Fox, T. Bilgic,(1996). A Requirement Ontology for Engineering Design, in: *Proceedings of the Third International Conference on Concurrent Engineering*, pp. 343-351.
- Ahmed, S., Kim, S., Wallace, K. M., (2005). A Methodology for Creating Ontologies for Engineering Design, In Proc. Of ASME DTM, 2005, DETC2005-84729.
- Yoshinobu Kitamura, Riichiro Mizoguchi,( 2004) Ontology-Based Functional-Knowledge Modeling Methodology and Its Deployment. *EKAW* : 99-115.
- John s gero and Udo kannengiesser. (2006) A function-behaviour-structure ontology of processes. *Design Computing and Cognition '06*. Part 6,407-422.

- Borst, P. Akkermans, H., and Top, J, (1997). Engineering Ontologies, Human-Computer Studies, 46(2/3), pp. 365-406.
- J.F. Sowa,(1984). Conceptual Structures: Information Processing in Mind and Machine, Addison-Wesley Publishing Company, Reading, MA.
- D. Genest, M. Chein, (2005).A Content-search Information Retrieval Process Based on Conceptual Graphs. Knowledge And Information Systems, vol. 8, no. 3, Springer, pp. 292–309.
- R. Dieng-Kuntz, D. Minier, M. Ruzicka, F. Corby,O. Corby, L. Alamarguy, (2006) .Building and using a medical ontology for knowledge management and cooperative work in a health care network, Computers in Biology and Medicine 36 (7-8) 871–892.
- O. Corby, R. Dieng-Kuntz, C. Faron-Zucker, F. Gandon,( 2006) . Searching the Semantic Web: approximate query processing based on ontologies, IEEE Intelligent Systems Journal 21 (1) 20-27.
- J.F. Baget, M.-L. Mugnier, (2002). Extensions of simple conceptual graphs: the complexity of rules and constraints, Journal of Artificial Intelligence Research (JAIR) 16 , 425–465.
- Abecker, A., Bernardi, A., Hinkelmann, K., K'uhn, O., & Sintek, M. (1998). Toward a technology for organizational memories. IEEE Intelligent Systems, 13(3), 40–48.
- CHEIN M. & MUGNIER M.-L. (1992). Conceptual Graphs : Fundamental Notions. Revue d'Intelligence Artificielle, 6(4), 365–406.
- Philippe Leblanc, (2008) Telelogic. Modélisation SysML par l'Exemple. Revue de génie logiciel. ISSN 1265-1397 , no 83.
- Ramesh, B., and Dhar, V. (1992), "Supporting Systems Development by Capturing Deliberations during Requirements Engineering", IEEE Transactions on Software Engineering, Vol. 18, No 6, pp. 498-510.
- Tautz, C.;Wangenheim, C.G.V. (1998), "REFSENO REpresentaion Formalism for Software Engenieing Ontologies", IESE-Report NO. o15.98/E, Version 1.1.
- Henninger, S. (1998), "An Environment for Reusing Software Process", In Proc. of the 5th IEEE International Conference on Software Reuse (ICSR5), Victoria, BC, Canada.
- Kim,Y., Kang, S., Lee, S.,&Yoo, S. (2001).A distributed, open, intelligent product data management system. International Journal of Computer Integrated Manufacturing, 14, 224–235.
- Gao, J. X., Aziz, F. L., Maropoulos, P. G., & Cheung, W. M. (2003). Application of product data management technologies for enterprise integration. International Journal of Computer Integrated Manufacturing, 16(7-8), 491–500.
- Kopena, J., & Regli,W. C. (2003). Functional modeling of engineering designs for the semantic web. IEEE Data Engineering Bulletin, 26(4), 55–61.