International Journal of Control and Automation
Vol.7, No.8 (2014), pp. 143-156
http://dx.doi.org/10.14257/ijca.2014.7.8.13

An Ontology-based Method for Process Modeling in Multi-Disciplinary Collaborative Design

Hao Yongping,*, Wang Chunyan, Zeng Pengfei and Shao Weiping

R&D Center of CAD/CAM Technology, Shenyang Ligong University, Shenyang 110159, China
yphsit@126.com

Abstract

Process modeling is an important task for collaborative design of a multi-disciplinary nature. This paper proposes an ontology-based method to model the collaborative design process. The aim is to effectively coordinate, allocate and share design resources. Concepts of task, sub-task, and local ontology in the collaborative design environment are defined. The tasks of achieving optimal resource allocations of design resources are divided into sub-tasks that are stored in the form of local ontology. Genetic algorithm is employed in match-researching among local ontology and different disciplines. Tasks and their discipline views are described in the ontology-enabled semantics-sharing of design resources in multi-disciplinary collaborative design. Process modeling of a missile is used as an example to demonstrate the capabilities of the proposed method and framework.

Keywords: *Multi-disciplinary collaborative design; Process modeling; Ontology; Task decomposition; Discipline view*

1. Introduction

Modern engineering design has become more and more multi-disciplinary and collaborative in nature [1-9]. A multi-disciplinary team often works together on a product relying on a collaborative mechanism that assists design task allocation management and collaboration [10-12]. The benefits of adopting such an engineering design paradigm includes reducing design cycle-time, decreasing design costs, improving design quality, and increasing product competitive advantages. In fact, Hazelrigg[13] maintains that engineering design should always be treated as a multi-disciplinary activity.

For a multi-disciplinary design team to function effectively, a number of issues need to be addressed. Designers with differing discipline backgrounds often converse with their own domain-specific terminology and conceptualization methods. Their ways of knowledge representation and process modeling are also different. As a consequence, researchers have suggested and developed different techniques for better understanding and collaboration among multi-disciplinary team members. The key is to respect the expertise of a domain export and meanwhile provide effective communication mechanisms for the designers to collaborate with each other in achieving a common objective. Failing to do this, multi-disciplinary design teams will not function as desired. The situation is further complicated by different point-of-views held by individuals of different disciplinary backgrounds over the same design process, and by different and often contradicting constraints. Yet, they have to be able to share the knowledge and resources, and agree to a set of common design goals.

ISSN: 2005-4297 IJCA

A number of researchers advocated that multi- disciplinary collaborative design be carried out with a common process modeling goal of achieving effective use of design resources. Such research is commonly termed Multi-disciplinary Design Optimization or MDO in short. For example, Amer *et al.*, [14] and Zhang *et al.*, Reference [15] proposed a method of transforming MDO into a multi-objective optimization problem. In other words, a possible infinite set is found for a multi-objective problem in which improvement over one objective can only occur when another objective is penalized. A large amount of literature has reported adoption of the method [4, 5, 6, 8, 9, 14, 15, 16, 17]. Although these optimization methods may appear to have utilized the design resources, they ignore the difference between multi-objective optimization and multi-disciplinary optimization. In fact, not all discipline- oriented problems can be expressed as problems with optimization objectives.

Process modeling methods have been recently studied by a number of researchers with a goal of optimally allocating and sharing design resources. For optimally allocating design resources, many adopted a multi- disciplinary design optimization method. For design resources sharing, use of ontology seems to be effective. This is because the knowledge in ontology representation can clearly and formally express concepts and relationships among them [18]. This paper focuses on multi-disciplinary and collaborative product design problems. Ontology modeling has been proposed, as ontology is a capable modeling tool at semantic and knowledge levels [16-26]. We first defined three task-decomposition rules: decomposition-granularity moderating principle, task- amount balancing principle, and completion-time consistency principle. According to the task-decomposition rules, a design task is divided into several sub-tasks. In doing so, product design becomes a more and more transparent and specific process. There are two main phases in this approach. The first one is decomposition of a design task into sub-tasks according to the task-decomposition rules. The second one is to match-search between local ontology and disciplines. In this respect, three steps are invovled. Firstly, sub-tasks are mapped into local ontology. Secondly, all the local ontology is stored and edited using protégé 4.0.2 [27] which can explicitly express concepts and relationships of a product. It can also avoid semantic heterogeneity as much as possible and support reasoning with its Reasoner FaCT++. Thirdly, since more than one discipline-combination may exist for one sub-task design, matching between local ontology and disciplines is sought by using a genetic algorithm. A mathematic model for match-searching has been developed and used to obtain accurate search results. A case study has been conducted to demonstrate its functionalities.

2. Process Modelling for Multi-Disciplinary Collaborative Design

2.1. Collaborative Design Task

A collaborative design task is defined as a series of design modules that are needed to carry out the design task. In order to achieve an optimal allocation and sharing of design resources, a Task needs different designers to work on in a collaborative way. Task is defined as a four-tuple,

$$\text{Task}=(\text{Tname}, \text{Person-List}, \text{Resource}, \{\text{Sub-Task}\}) \qquad (1)$$

where, Tname─the name of the design task; it is the only identifier for the task;
Person-List─the list of designers involved, which is formally defined as a triple,
Person-List=(name,professional-expertise,department);

Resource─a set of design resources, *i.e.*,
Resource=(List of resources required, List of resources available);
{Sub-Task}─a set of sub-tasks defined as,
{Sub-Task}={Sub-Task1, Sub-Task2, …, Sub-Taski, ..., Sub-Taskn}

2.2. Sub Collaborative Design Task

A Sub-task in collaborative design corresponds to certain characteristics of a design task. We define a Sub-task as a six-tuple,

Sub-Task=(STName, P, M, R, C, Information) (2)

where, STName─the name of sub-tasks;

P─a property set, *i.e.*,

P=<PType, PName, Pvalue>, PType, PName and Pvalue denote attribute type, name and value, respectively.

M─a method set including approaches used in achieving the sub-task.

R─a set of relationships between different sub-tasks. The relationships can be of an inheritance, instance, property, incompatible, and/or connection type.

C─a set of constraints such as number constraint, inherent constraint, and experience constraint. A number constraint always appears as a number, *e.g.*, a number constraint of design sizes. An inherent constraint defines the inheritance relationships among sub-tasks and these relationships do not change. An experience constraint is a series of design experiences that design experts have in the course of product design. Experience constraint is essential in product design, and it is often intangible.

Information─interpretations of design reference of sub-tasks. In other words, it is a further description of the sub-tasks.

2.3. Task-decomposition

For any complex and large-scale design product, it is essential the design task be decomposed into sub-tasks so as to improve design efficiency and success. Sub-tasks from task-decomposition can facilitate design collaboration. Process modelling can also be made easier. Design efficiency can largely be affected by the way the tasks are decomposed. Therefore, some decomposition-rules need to be defined and used to ensure that a design task is decomposed properly. Given any sub-task sets, $\{Sub\text{-}Task\}_i$ (i=1, 2, …, n) as the results of task-decomposition, the following always holds

$$\cup \{Sub\text{-}Task\}_i = Task \ (i = , 2, \ldots, n) \qquad (3)$$

This is equivalent to say that the logical union of any set of sub-tasks from task-decomposition is always equal to the design task. Therefore, three decomposition-rules are defined. The three rules can be translated into constraints for calculation purposes.

Rule 1: Principle of Appropriate Decomposition Granularity. The aim of task-decomposition is to simplify a complex collaborate design. Hence, decomposition granularity cannot be too big. It cannot be too small, either. Otherwise, heavy overlapping between multi-disciplines will occur. The principle of appropriate decomposition granularity is formally expressed as follows.

Assuming $G(x)$ is a measure function for decomposition-granularity, in a domain of $x \in \{Sub\text{-}Task_1, Sub\text{-}Task_2, \ldots, Sub\text{-}Task_i, \ldots, Sub\text{-}Task_n\}$, the principle of appropriate decomposition granularity is expressed as the following constraints.

$$\forall i, \left\| G\,(Sub \quad Task_{\,i}) \right\| \le \left| g_0 \right| \tag{4}$$

Where, g_0─boundary value.

In other words, decomposition granularity should be kept under a given value, and any task-decomposition is deemed invalid if its granularity is either too larger or too smaller than the given boundary value, g_0.

Rule 2: Principle of equivalent task amount. The principle of equivalent task amount entails that every sub-task will have its local ontology in the local ontology definitions. This way, repetition of creating one or more local ontology is avoided. This also means that the number of local ontology will be the same as the total number of sub-tasks (n).

Given any two sub-tasks in a set, *Sub-Task*$_i$ and *Sub-Task*$_j$, the task amount function $\mu(x)$ applies across the domain, $x \in \{Sub\text{-}Task_1, Sub\text{-}Task_2, \ldots, Sub\text{-}Task_i, \ldots, Sub\text{-}Task_n\}$. Based on the task amount equivalence principle, every sub-task in the set should satisfied,

$$[\mu]_{\min} \le \left\| \left\| \mu(Sub - Task_{\,i}) \right\| - \left\| \mu(Sub - Task_{\,j}) \right\| \right\| \le [\mu]_{\max}, \forall\, i,\, j \in [1, n],\, i \ne j\,. \tag{5}$$

μ_{min} and μ_{max} are the minimum and maximum of task-amount, respectively. They are based on designer's experiences, independent of the functions of sub-tasks.

Rule 3: Principle of consistent completion time. The principle of consistent completion time is another importance measure for task-decomposition. Completion time of each sub-task affects that of the entire product design. Therefore, completion time for all sub-tasks should be kept as similar as possible so as to maximise design efficiency.

Assuming *t(x)* as a function of completion-time for a sub-task in the domain of $x \in \{Sub\text{-}Task_1, Sub\text{-}Task_2, \ldots, Sub\text{-}Task_i, \ldots, Sub\text{-}Task_n\}$, the completion time consistency principle is represented as

$$\left| t(Sub \quad Task_{\,i}) \quad t(Sub \quad Task_{\,j}) \right| \le \Delta t_{\max} \qquad \forall i,\, j \in n \tag{6}$$

where Δt_{max} represents the maximum completion time difference between two sub-tasks.

2.4. Process Modeling in Multi-Disciplinary Collaborative Design

Process modeling in multi-disciplinary collaborative design requires optimal allocation and sharing of resources. To optimize resource allocation, every design task is decomposed based on decomposed-rules defined as above, and the decomposed sub-tasks are stored by local ontology. Match-search between local ontology and disciplines is then carried out. In order to enable better sharing of knowledge of different disciplines, ontology modeling is used, by which knowledge from multiple disciplines is expressed in a uniform format, and communications and sharing among different disciplines are facilitated. Therefore, knowledge reuse becomes easier when semantics sharing is achieved for design resources in a multi-disciplinary collaborative design environment. Since more than one discipline is needed in a sub-task design process, different disciplinary designers will need to collaborate with each other for the sub-task. To achieve a seamless connection between knowledge from varying design resources, it is essential that each sub-task get its best discipline-combination.

3. Ontology-driven Process Modeling

Ontology is a philosophical discipline, a branch of metaphysics that deals with the nature of being. The concept of ontology has attracted a good deal of attention, partly due to the growing need, to develop terminology for building knowledge bases for a specific domain in a way that can be understood by a machine.

Designers from different disciplines may have differing semantic understanding over the same issue. To solve this problem, ontology language (*i.e.*, W3C's recommendation Web Ontology Language-OWL [28]) can be used to describe sub-tasks. With such a description, semantic heterogeneity between design resources can be overcome; designers of different disciplines can better understand each other; seamless docking of resources can be achieved; and a real sense of sharing of entire design resources can be had. Stored by the protégé ontology editor that provides all the modeling functionalities needed to create ontology compliant with OWL, the local ontology for the corresponding sub-tasks can be defined as a four-tuple.

$$\textit{Local Ontology} = [\textit{Name}, \textit{Concept}, \textit{Property}, \textit{Relation}] \tag{7}$$

where, *Name*—the identifier of the local ontology,
concept—a set of concepts for the local ontology,
property—a set of attributes for the local ontology,
relation—a set of relationships among this local ontology.

All the local ontology is edited and stored using protégé, which enables the concepts and relationships among the products to be explicitly expressed. Ontology mapping is used to achieve conversions between sub-tasks and local ontology, which is effectively a process for data storage to convert sub-tasks into their corresponding local ontology. As a result, local ontology is obtained from a sub-task which is stored in the ontology format by using the ontology description language and ontology edit tools. The main objective of this conversion process is to change concepts and relationships between them in a sub-task into classes or properties in local ontology. Mapping may be carried out in the following three different ways:

Mapping 1: Mapping from sub-tasks stored by the structure description to their local ontology. Tables of the relational database that store sub-tasks are mapped as concepts of a class in local ontology; fields can be viewed as concepts of class properties. Furthermore, terms of local ontology should be kept the same as those in the tables and fields. Figure 1 shows an example of a missile product whose part data stored in a relational database is mapped to an ontology format.

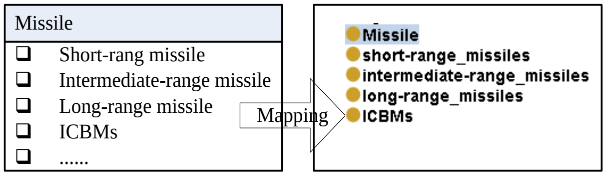

Figure 1. An Example of Mapping 1

Mapping 2: Mapping from sub-tasks stored via a semi-structured description to their local ontology

Before mapping takes place, the semi-structured descriptions for sub-tasks are abstracted and presented in UML class diagrams, which generate their local ontology automatically [29, 30]. Figure 2 describes the flow of the mapping process.

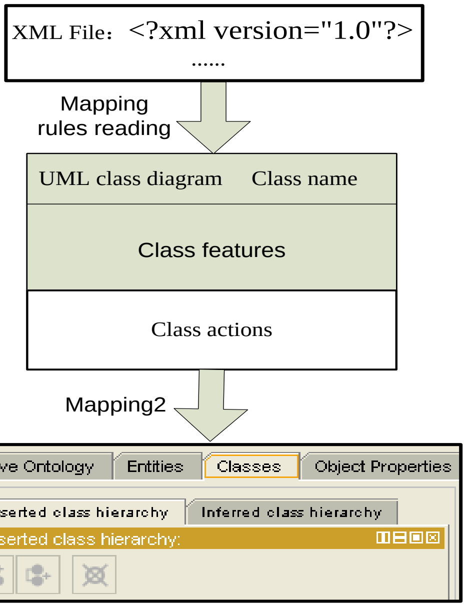

Figure 2. Process of Mapping 2

Mapping 3: Mapping from sub-tasks which are stored by unstructured descriptions to their local ontology. The mapping process converts this type of sub-tasks into classes or constraints of attributes in Mapping 1 or Mapping 2.

Upon implementing the three mappings, all sub-tasks will be converted into their local ontology. Therefore, the job of defining a sub-task becomes that of converting it to its local ontology. In order to ensure quality design in a collaborative environment, further reasoning support is needed to verify whether local ontology can replace sub-tasks for the design project. For this, we used Reasoner FaCT++ which can check whether or not all the statements and definitions in local ontology are mutually consistent. Reasoner FaCT++ can also help maintain valid hierarchical descriptions. Through mappings, all local ontology is obtained from sub-tasks that are from the local ontology layer. This is shown in Figure 3.

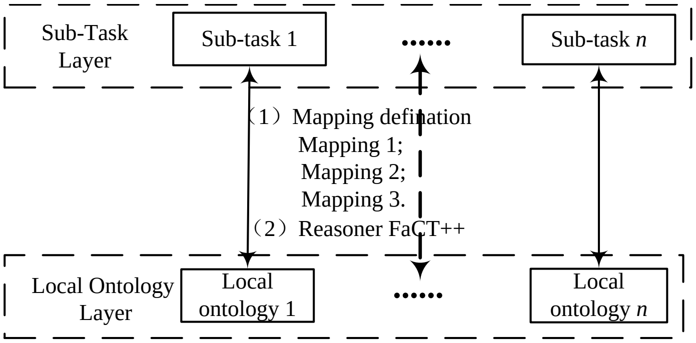

Figure 3. Local Ontology Layer

4. Match-searching Between Local Ontology and Disciplines

The concept of Discipline view (*DV*) is introduced to facilitate knowledge sharing across different disciplines. Match-searching is carried out based on a genetic algorithm (GA).

4.1. Discipline View

In the lifecycle of product development, professional knowledge is often expressed and categorized into disciplines. Therefore, multiple disciplines are to be considered when the knowledge being required for product design cuts across different disciplines. For example, when designing the chassis of a car, multi-disciplinary knowledge such as mechanical, materials and mathematical disciplines is needed.

Different disciplines often have their domain knowledge represented in their own ways, *e.g.*, concepts and terminology. This imposes a problem in sharing the knowledge across different disciplines. Discipline view is introduced to overcome this problem. *DV* represents an observation of local ontology from its own perspective; it also represents mapping of local ontology of different disciplines. Meanwhile, *DV* is a description of the characteristics of local ontology from the perspective of the discipline.

For the sake of parameterization and standardization of design knowledge, and quantifying the knowledge for computing purposes, *DV* is defined as a four-tuple.

$$DV_{ij}=[\mathit{No.}\ ,\ \mathit{Performance\text{-}decomposition},\ \mathit{Performance\text{-}element\text{-}set},\ \mathit{Design\text{-}function}] \tag{8}$$

where, *No.*—serial number of *DV*, *e.g.*, *No*. is *ij*, when DV_{ij} presents the discipline view of local ontology *i* to discipline *j*;

Performance-decomposition—a refining process of the design performance;

Performance-element-set—a set for elements from *performance-decomposition*;

Design-function—an explicit representation of constraints for elements from the *performance-element-set*.

Use of the discipline view definition helps consolidate knowledge from different disciplines, hence facilitates multi-disciplinary, collaborative product development process. Knowledge can be shared between multiple disciplines, and this is exactly the aim of process modelling for multi-disciplinary product design.

4.2. Match-searching

Match-searching needs to be expressed mathematically. For this reason, genetic algorithm has been used, which can provide more accurate search result.

4.2.1. Mathematical Model for Match-Searching: More often than not, more than one disciplines (or a combination of disciplines) are present in a design task or a sub-task. Figure 4 shows an example of a sub-task that can be achieved by four different discipline-combinations. In this case, effort is needed to select the best set of combination in order to achieve better utilization of design knowledge and resources.

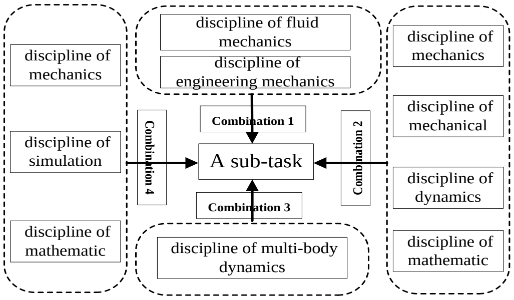

Figure 4. A sub-task with its Four Different Discipline-Combinations

For product *P*, its design task can be decomposed into a number of sub-tasks, $Sub\text{-}Task_i$ ($i \in [1, n]$, *n* – total number of sub-tasks). There are a number of disciplines (*mi*) that are needed for doing the design task, *mi* ($1 \leq mi \leq m$, *m* – total number of available disciplines). Therefore, the mathematical model of the match-searching can be expressed as

$$\max \quad f = \sum_{i=1}^{i=n} Pi \quad (\sum_{j \in n} W_{ij} \quad DV_{ij}) \tag{9}$$

$$s.t. \begin{cases} \sum_{i=1}^{n} Pi = 1; \\ \sum_{j \in n} W_{ij} = 1, (\forall i \in [1, n]); \\ 0 < Pi < 1, i \in [1, n]; \\ 0 < W_{ij} < 1, i \in [1, n], \ j \in [1, m]; \\ DV_{ij} \equiv 1, \text{when discipline } j \text{ can collaborate to complete sub-task } i; \\ DV_{ij} \equiv 0, \text{when discipline } j \text{ can not collaborate to complete sub-task } i. \end{cases}$$

Parameter *Pi* reflects the significance level of a local ontology compared with the rest of local ontology. W_{ij} is a matching coefficient between discipline *j* and sub-task *i*. Different calculation methods are to be used for different matching coefficients. For collaborative design, there exist at least two sub-tasks in task decomposition, hence the constraint, $0<Pi<1$. At the same time, there are also at least two disciplines involved in achieving each sub-task, hence the constraint, $0<W_{ij}<1$. As can be seen from the above model, choosing the best discipline combination to do a sub-task has now becoming an optimization problem.

4.2.2 Realization of Match-Searching: Genetic algorithm (GA) is an intelligent search method requiring domain-specific knowledge to solve a problem. By mimicking the evolutionary process of nature, such algorithms have been employed as global search and optimization techniques for various scientific and engineering problems. It is an ideal method for the realization of match-searching between local ontology and different disciplines. Figure 5 shows such a process–a process of getting the best discipline-combination match for each sub-task.

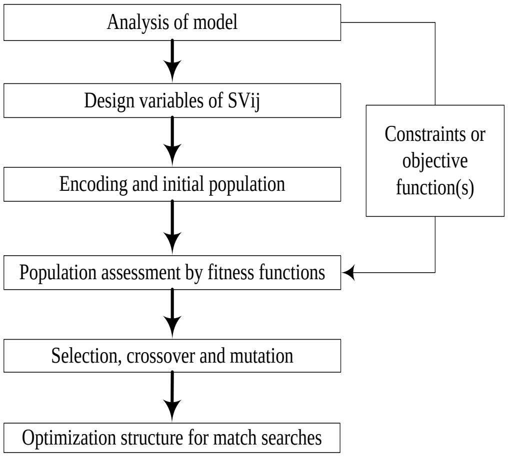

Figure 5. Genetic Algorithm used for Discipline-Combination Match-Searching 0/1 Encoding is used in GA and the Encoding Method is explained as Follows

- Similarity Matrix

The elements (denoted as W_{ij}) in the similarity matrix represent similarities between all the local ontology and multiple disciplines, i.e. W_{ij} denotes the level of similarity. The more knowledge for discipline j is used for designing local ontology i, the lager the value of W_{ij} Conversely, the less knowledge for discipline j is used for designing local ontology i, the smaller the value of W_{ij} A similarity matrix for local ontology (n) and disciplines (m) are shown in Table 1.

Table 1. Similarity Matrix

L \ *D*	*1*	*2*	...	*j*	...	*m*
1	W_{11}	W_{12}	...	W_{1j}	...	W_{1m}
2	W_{21}	W_{22}	...	W_{2i}	...	W_{2m}
…	...	...	...	...	...	...
i	W_{i1}	W_{i2}	...	W_{ij}	...	W_{im}
…	...	...	...	...	...	...
n	W_{n1}	W_{n2}	...	W_{nj}	...	W_{nm}

L: local ontology; D: discipline; W_{ij}: (0, 1)

- Correlation matrix

The elements (denoted as Sgn_{ij}) in the correlation matrix (Table 2) are assigned following a specific rule. That is, Sgn_{ij} is assigned to 1 if W_{ij} from the similarity matrix is bigger or equal to the given threshold value (F); otherwise it is assigned to 0.

Table 2. Correlation Matrix

L \ *D*	*1*	*2*	...	*j*	...	*m*
1	Sgn_{11}	Sgn_{12}	...	Sgn_{1j}	...	Sgn_{1m}
2	Sgn_{21}	Sgn_{22}	...	Sgn_{2j}	...	Sgn_{2m}
…	...	...	...	...	...	...
i	Sgn_{i1}	Sgn_{i2}	...	Sgn_{ij}	...	Sgn_{im}
…	...	...	...	...	...	...
n	Sgn_{n1}	Sgn_{n2}	...	Sgn_{nj}	...	Sgn_{nm}

- Initial population of GA search

Sgn_{ij} in the correlation matrix is set to 1 if discipline j is used to finish sub-task i; it is set to 0 if discipline j is not used for completing sub-task i. For a sub-task i, its initial population of GA search then becomes, $[Sgn_{i1}, Sgn_{i2}, ..., Sgn_{ij}, ..., Sgn_{im}]$.

Understandably, design experience also plays an important role in the initial population setting. Fitness function ($F(f(x))$) of the population is a function of design constraints or objective function; although it is more of an objective function. Through GA-based match-searching, each sub-task will be designed using its best-fit discipline-combination. Therefore, design resources are utilised and allocated in an efficient way.

5. An Ontology-Driven Process Modeling Framework

In the framework, there are two key phases in modeling the ontology-driven process for multi-disciplinary collaborative design. The first one is the division of a design task into sub-tasks in accordance with task-decomposition rules defined previously. The other is the search for the best discipline-combination for each sub-task through a GA-based match-searching algorithm. The steps involved in the ontology-driven process modeling framework are shown in Figure 6 and also explained as follows.

Step 1: Given a product to be designed, a hierarchical description is firstly conceived through databases available.

Step 2: Based on the hierarchical description, a conceptual model of the product is formed in terms of a (design) task.

Step 3: Following the task-decomposition rules, the design task is divided into several sub-tasks forming the sub-task layer.

Step 4: Each sub-task is mapped to its local ontology through the three types of mapping methods (as defined in Section 2). This is a one-to-one mapping.

Step 5: Number and sequence all disciplines. The more often a discipline is used, the further ahead it should be sequenced in the entire discipline list.

Step 6: In order to optimize all the sub-tasks in form of local ontology, mathematical models of match-searching between local ontology and disciplines are defined.

Step 7*:* Through GA-based match-searching, the best discipline-combination for each sub-task is obtained, which effectively gives the best discipline-combination for achieving the design task.

Step 8: With the unified discipline view, each local ontology gets its own discipline view corresponding to the obtained best discipline-combination. That is, a discipline view is a view from local ontology to its best discipline-combination. For an instance, if discipline 1, 3 and *j* are needed to complete design sub-task *i*, discipline views DV_{i1}, DV_{i3} and DV_{ij} will be obtained.

Step 9: Based on the discipline view, design data of all local ontology is stored.

Step 10: Product design is completed once all design data is integrated seamlessly.

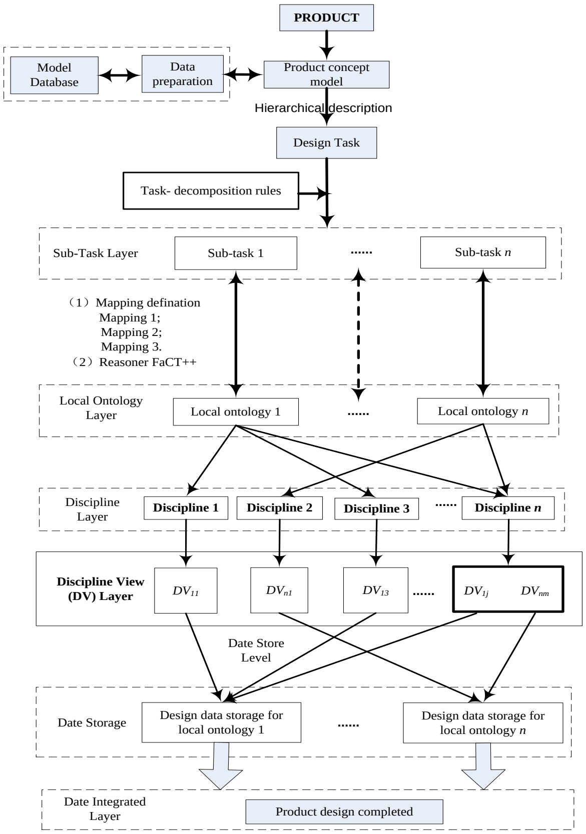

Figure 6. The Ontology-Driven Process Modelling Framework

6. Case Study

The proposed framework has been implemented. This section provides a case study based on the framework. The example is a missile product. The design process (as shown in Figure 6) undergoes two phases, division of the design task into sub-tasks in accordance with task-decomposition rules, and search for the best discipline-combination for each sub-task through a GA-based match-searching algorithm. Local ontology is stored and edited using ontology edit tool, protégé 4.0.2. Some of the key steps in the process are expressed as follows.

Step 1: Data preparation and hierarchical description for the missile product.

Step 2: Based on the hierarchical knowledge description for the control system, navigation system, and sensors, the design task is formulated.

Step 3: The design task is divided into several sub-tasks according to the task-decomposition rules.

Step 4: Converting each sub-task into local ontology (stored and edited using protolo). This is achieved through the mapping rules defined in the framework. Local ontology for design tasks including sensors and control system can be seen in the middle section of Figure 7.

Step 5: Eight disciplines are involved in the design task, *e.g.*, materials, control, electronics and hydraulics. They are selected based on the prior design experiences with regard to the product. They are also numbered properly. For example, the discipline of materials is considered the most commonly used one and therefore marked as number 1.

Step 6: Mathematical models of match-searching between local ontology and disciplines are set up. Note that the model constraints are not included herein.

Step 7: Construction of the similarity matrix: Take material design of the control system as an example, its similarity matrix is shown in Table 3.

Table 3. Example of a Similarity Matrix

Discipline	1	2	3	4	5	6	7	8
Materials	0.56	0	0	0.13	0.07	0.0005	0.132	0.1003

Step 8: Construction of the correlation matrix: With a threshold value of 0.008, its correlation matrix is shown Table 4.

Table 4. Example of a Correlation Matrix

Discipline	1	2	3	4	5	6	7	8
Materials	1	0	0	1	1	0	1	1

Step 9: Table 4 gives an initial population, [1, 0, 0, 1, 1, 0, 1, 1]. If the threshold is set to be 0.8 and mutation probability 0.001, the results of match-searching for the local ontology of material design in the control system are disciplines of materials, control, electronics and hydraulics, with the algorithm terminating after 200 iterations.

Step 10: Determination of discipline-views: For material t, the discipline-views are DV_{t1}, DV_{t4}, DV_{t5}, and DV_{t8}.

Step 11: Data storage.

Step 12: Missile design completion.

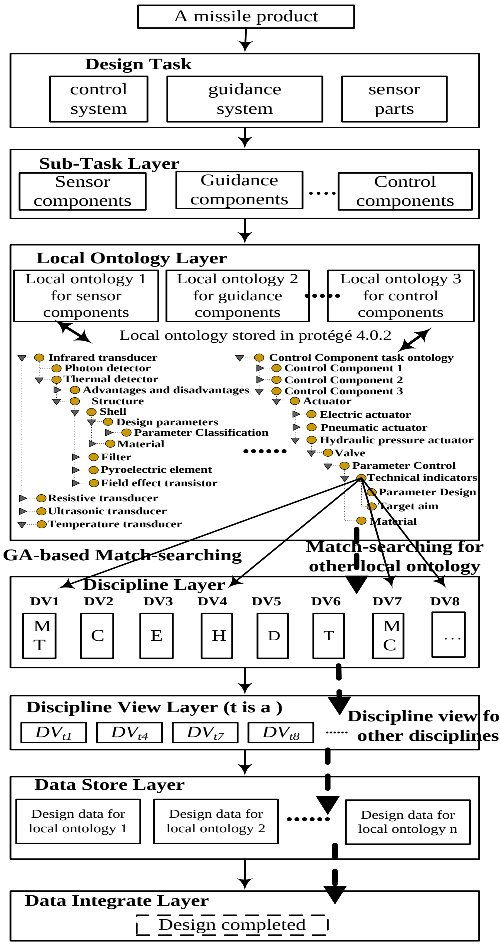

Figure 7. Process Flow for a Missile Design

7. Conclusions

In this study, a process modeling method for multi-disciplinary collaborative design has been proposed. The modeling method is ontology-based. Three task-decomposition rules are defined and successfully implemented. Implementation of the task-decomposition rules divides a design task into sub-tasks. This way, product design process becomes more transparent and targeted. This method also enables optimal resource allocations and resource sharing. Accordingly, a two-phase approach has been developed. The first one is the division of a design task into sub-tasks in accordance with task-decomposition rules defined previously. The other is the search for the best discipline-combination for each sub-task through a GA-based match-searching algorithm. For the latter, three steps are included. First, sub-tasks are mapped into local ontology by applying the mapping rules defined in the study. Secondly, all local ontology is stored and edited using protégé 4.0.2. This way, concepts and relationships in the product design process are expressed explicitly and semantic heterogeneity is reduced. Reasoning with its Reasoner FaCT++ is provided. Thirdly, given that not only one discipline-combination may exist for one sub-task, GA is used for match-searching between local ontology and disciplines to obtain more accurate search results. The framework has been implemented and a case study has been conducted to demonstrate the methodology.

The proposed ontology-based modeling method still has some limitations. Most of the ontology storing work is still largely manually done. The framework thus far only deals with one specific area; more areas of applications need to be considered to take full advantage of the framework. Future work is also needed to consider the impact among different design resources.

References

[1] J.-H. Park and K.-K. Seo, "A knowledge-based approximate life cycle assessment system for evaluating environmental impacts of product design alternatives in a collaborative design environment", Advanced Engineering Informatics, vol. 20, no. 2, **(2006)**, pp. 148-154.

[2] E. Arias, H. Eden, G. Fischer, A. Gorman and E. Scharff, "Transcending the individual human mind-creating shared understanding through collaborative design", ACM Transactions on Computer-Human Interaction, vol. 7, no. 1, **(2000)**, pp. 84-113.

[3] L. M. Camarinha-Matos and H. Afsarmanesh, "Collaborative networks: a new scientific discipline", Journal of Intelligent Manufacturing, vol. 1, no. 4-5, **(2005)**, pp. 439-452.

[4] N. Cross, "Designedly ways of knowing: design discipline versus design science", Design Issues, vol. 17, no. 3, (**2001),** pp. 49-55.

[5] Y. Q. Di, B. H. Li, X .D. Chai and P. Wang, "Research on collaborative modeling & simulation platform for multi-disciplinary virtual prototype and its key technology", Computer Integrated Manufacturing System (in Chinese), vol. 11, no. 7, **(2005)**, pp. 01-908.

[6] M. A. Rosenman, G. Smith, M. L. Maher, L. Ding and D. Marchant, "Multidisciplinary collaborative design in virtual environments", Automation in Construction, vol. 16, no. 1, **(2007)**, pp. 37-44.

[7] L. Chen and G. D. Jin, "Project management of web-based product development teleservice", International Journal of Advanced Manufacturing Technology, vol. 32, no. 1-2, **(2007)**, pp. 162-168.

[8] W. O'Brien, L. Soibelman and G. Elvin, "Collaborative design processes: An active- and reflective-learning course in multidisciplinary collaboration", Journal of Construction Education, vol. 8, no. 2, **(2003)**, pp. 78-93.

[9] H. K. Lin and J. A. Harding, "A manufacturing system engineering ontology model on the semantic web for inter-enterprise collaboration", Computers in Industry, vol. 58, no. 5, **(2007)**, pp. 428-437.

[10] L. H. Wang, W. M. Shen, H. L. Xie, J. Neelamkavil and A. Pardasani, "Collaborative conceptual design-state of the art and future trends", Computer-Aided Design, vol. 34, no. 13, **(2002)**, pp. 981-996.

[11] M. Rosenman and F. J. Wang, "A component agent based open CAD system for collaborative design", Automation in Construction, vol. 10, **(2001)**, pp. 383-397.

[12] X. F. Zha and H. Du, "Knowledge-intensive collaborative design modeling and support Part I: Review", distributed models and framework, Computers in Industry, vol. 57, no. 1, **(2006)**, pp. 39-55.

[13] X. F. Zha and H. Du, "Knowledge-intensive collaborative design modeling and support", Part II, System implementation and application, Computers in Industry, vol. 57, no. 1, **(2006)**, pp. 56-71.

[14] G. A. Hazelrigg, "Systems Engineering: An Approach to Information-Based Design", Prentice Hall, Upper Saddle River, NJ, **(1996)**.

[15] F. R. Amer, L.-S. He and Z. Qasim, "Multidisciplinary design and optimization of satellite launch vehicle using latin hypercube design of experiments", Computer Aided Drafting, Design and Manufacturing, vol. 19, no. 1, **(2009)**, pp. 1-7.

[16] J. Zhang, Y.-J. Liu and B.-L. Li, "Multidisciplinary design optimization of robot system based on collaborative optimization", Machine Design and Research (in Chinese), vol. 24, no. 5, **(2008)**, pp. 235-237.

[17] M. H. Han and J. T. Deng, "Improvement of collaborative optimization", Chinese Journal of Mechanical Engineering (in Chinese), vol. 42, no. 11, **(2006)**, pp. 34-38.

[18] L. X. Gu and C. L. Gong, "A comparison of multidisciplinary design optimization methods", Journal of Projectiles, Rockets, Missiles and Guidance (in Chinese), vol. 25, no. 1, **(2005)**, pp. 60-62.

[19] W. Y. Zhang and J. W. Yin, "Exploring semantic web technologies for ontology-based modeling in collaborative engineering design", International Journal of Advanced Manufacturing Technology, vol. 36, no. 9-10, **(2008)**, pp. 833-843.

[20] B. David, J. Gao, K. Case, J. Harding, B. Young, S. Cochrane and S. Dani, "A Framework to integrate design knowledge reuse and requirements management in engineering design", Robotics and Computer-Integrated Manufacturing, vol. 24, no. 4, **(2008)**, pp. 585-593.

[21] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman and J. Widom, "The TSIMMIS Project: Integration of Heterogeneous Information Sources", Information Processing Society of Japan (IPSJ 1994), Tokyo, Japan, **(1994)** October.

[22] H.-K. Lin, J. A. Harding and M. Shahbaz, "Manufacturing system engineering ontology for semantic interoperability across extended project teams", International Journal of Production Research, vol. 42, no. 24, **(2004)**, pp. 5099-5118.
[23] G. Thomas, "Toward principles for the design of ontologies used for knowledge sharing", International Journal Human-Computer Studies, vol. 43, no. 5-6, **(1995)**, pp. 907-928.
[24] W. N. Borst, "Construction of engineering ontologies for knowledge sharing and reuse", PhD thesis, University of Twente, **(1997)**.
[25] D. Lewis, K. Feeney, T. Tiropanis and S. Courtenage, "An active, ontology-driven network service for Internet collaboration", In: Proceedings of the ECAI 2004 Workshop on Application of Semantic Web Technologies to Web Communities, Valencia, Spain, **(2004)**.
[26] T. A. S. C. Vieira, M. A. Casanova and L.G. Ferrao, "An ontology-driven architecture for flexible workflow execution", WebMedia & LA-Web 2004 Joint Conference 10th Brazilian Symposium on Multimedia and the Web 2nd Latin American Web Congress, Ribeirão Preto, São Paulo, Brazil, **(2004)**, pp. 70-77.
[27] P. Raúl, H. Peter, C. Oscar, G.-P Asunción and J. Qiu, "An editorial workflow approach for collaborative ontology development", Lecture Notes in Computer Science, vol. 5367, **(2008)**, pp. 227-241.
[28] H. Ivan, "Web Ontology Language (OWL)", http://www.w3.org/2004/owl/, (accessed on 12.11.10), **(2010)**.
[29] M. Horridge, M., H. Knublauch, A. Rector, R. Stevens and C. Wroe, "A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools", University of Manchester, **(2004)**.
[30] Yan and L. Li, "Mapping from UML class diagram to ontology", Science Technology and Engineering (in Chinese), vol. 8, no. 13, **(2008)**, pp. 3645-3648.
[31] Z.-M. Xu, H.-J. Gu, Y.-Y. Ni and M.-Q. Zhu, "Design and implementation of a UML-to-OWL conversion tool", Journal of Hohai University (Natural Sciences) (in Chinese), vol. 35, no. 4, **(2007)**, pp. 478-482.

Authors

Hao Yongping, born in 1960, is currently a Professor and PhD at Shenyang Ligong University, China. His research interests include CAD/CAM and collaborative product design.

Tel: +86-24-24681234; E-mail: yphsit@126.com

Wang Chunyan, born in 1987, is currently a master candidate at Shenyang Ligong University, China.

Zeng Pengfei, born in 1978, is currently assistant Professor and PhD at R&D Center of CAD/CAM Technology, Shenyang Ligong University, China.

Shao Weiping, born in 1968, is currently a Professor and PhD at R&D Center of CAD/CAM Technology, Shenyang Ligong University, China.