13 ONTOLOGY SUPPORTED ADAPTIVE USER INTERFACES FOR STRUCTURAL CAD DESIGN

Carlos Toro[1], Maite Termenón[1], Jorge Posada[1],
Joaquín Oyarzun[2], Juanjo Falcón[3].

1. VICOMTech Research Centre,
{ctoro, mtermenon, jposada } @ vicomtech.es
2. LANIK S.A.
joyarzun@lanik.com
3. SOME S.L.
jfalcon@somesi.com

In the early stages of the Steel Detailing Design process (Structural Design), most of the activities are focused in the designer. Nowadays Detailing CAD packages offer a wide range of options that in some cases exceeds the ones needed to fulfil a specific task. Sometimes having such a wide range span can be self-defeating for a smooth process evolution as the designer has to browse repetitively in the user interface for a particular tool. In this paper we present a Knowledge based approach for the exploitation of semantic aspects (e.g. user intentions and tasks) for the real time automatic generation of graphical user interfaces on a Steel Detailing CAD software. We base our approach in international standards (CIS/2) for the specific domain and as test case we present a system implementation of the proposed schema.

1. INTRODUCTION

In the Steel Detailing process (Structural Design), most of the early activities are focused in the designer [2]. Actual Computer Aided Design tools (CAD) represent a time saving aid as they help in the 3D design process while, in the meantime, the contained information in the model can be used in other stages of the structure's life cycle. Today's Steel Detailing CAD software packages offer a wide span of options in order to perform myriads of manipulations on the elements contained in the model. In a sense, it can be said that the amount of options offered to a CAD user nowadays is as huge as the capabilities of the program itself.

It is common that the user has to navigate through the menus searching for a specific tool, slowing the design process and therefore falling into a situation where the cost of the design stage increases.

In this paper, we present a Knowledge based approach for the exploitation of semantic aspects (e.g. user intentions and tasks) for the real time automatic generation of graphical User Interfaces (UI) on a Steel Detailing CAD software. We

base our approach in international standards (CIS/2) for the specific domain and, as test case, we present a system implementation of the proposed schema.

This paper is organized as follows: In chapter two, we present a brief state of the art, mentioning some approaches relevant to our work. In chapter three we explain our methodology for the real time automatic generation of UI based on semantic technologies and we show, in chapter four, a test case based in our methodology, applied in the domain of Structural Design. Lastly, in chapter five we present some conclusions derived from the paper and future work.

2. RELATED WORK

In recent times, the problem of producing a good UI has been a constant topic in research papers and conferences. To highlight some approaches, we can mention the work by Furtado [4] who presented an ontology based methodology to produce UI where multiple users carry out multiple tasks in a universal context. In this approach, they divided the UI design in three layers: a conceptual layer where a domain expert defines an ontology of concepts, relationships and attributes of the domain; a logical layer where a designer specifies multiple models based on the ontology and a physical layer where a developer derives multiple user interfaces from the previously specified models with alternatives. The mentioned approach is not fully automatic as it needs a developer to take care of the alternatives programmatically. The interface development is focused in the modeling of the user which is opposite to our approach where we intend to model a manufacturing process of a specific-well known and standardized domain (Steel Detailing design).

Puerta [13] based his approach on a model ontology, showing a UI development environment and system implementation with both, static and dynamic behaviors. In this approach, the domain model allowed the creation of interfaces for medical applications. In our approach, we model the user as the executioner of a design task, meaning that the design process is, in fact, divided in stages where the user takes actions that need tools in order to fulfill. Hovestadt [**7**] presented a prototypical implementation of a graphical UI intended for the architectural design process. In this approach, the interface integrates CAD-like object manipulation and navigation through large data sets oriented to help the user in the process of shifting directly from construction tasks to navigation tasks. In this approach, the main focus is to change the representation of the menus from the traditional pull down distribution to a hyperbolic shape.

In a non UI approach related to the exploitation of semantics in a CAD program, Ekholm [2] presented a work where, by means of the modelling of a sociotechnical system, the user, his role and design activities are specified. His approach claims to enhance the functionality of a building design software in the problem definition and analysis phases of the design.

Igarashi [8] presented a methodology to design UIs focused on visual thinking activities (creativity design). The basic idea is to make transparent interfaces so that the user can directly interact with the target visual representations without using menus and buttons in a sort of predictive approach. For the structural

design problem, it would be difficult to predict tools as the process itself is engineering based although some creativity is allowed.

3. SEMANTIC BASED GENERATION OF USER INTERFACES

We define a CAD tool as the means whereby some act for design, drafting, display or modify entities in a computer graphically oriented environment is accomplished.
In the field of steel detailing design, the structure life cycle can be viewed from a product point of view, where the object being designed passes through a series of stages where transformations can be distinguished. Our approach for the UI generation starts with the prerogative that every stage of a structure life cycle can be divided in sub processes. In figure 1 (Right) a design sub process schema is presented. In each sub process, a set of CAD tools can be applied by the user. Any CAD tool can be placed in more than one sub processes.
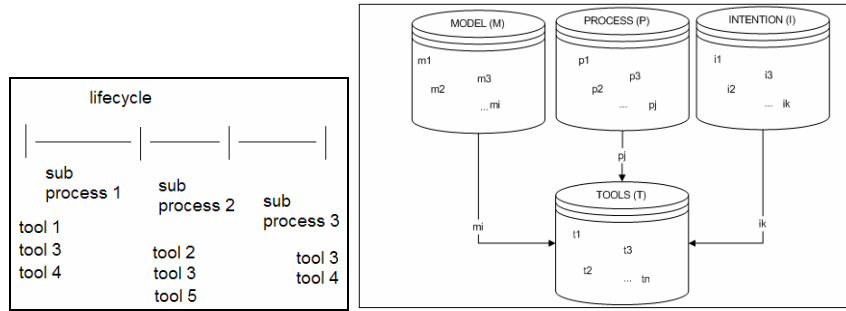

Figure 1 – (Right) Sub processes View, (Left) Model-Process-Intention View

We argue that an efficient UI should contain only the needed tools for a given sub process, model type and user intention. Upon the spectrum of techniques that can be used in order to model a process based application (e.g. relational databases, taxonomies, Petri Nets, etc), we choose to use an ontology approach in order to take advantage of the semantics inherited by the process modeling. To model the process ontology that allows the recommendation of the set of CAD tools via a query engine, we have used an international standardization effort (CIS/2) in order to name the classes and relations. The process ontology has a set of allowed tools (AT) with an object type property that allows the connection between them and the rest of the ontology. The set of tools is composed by actual tools belonging to the CAD system itself (e.g. element generation, primitives, queries, manipulators, etc), applets or extensions to the base system and even external calls to programs in the user's computer. For a given Model m_i, Process p_j and User Intention i_k, the goal will be to produce a set of recommended tools extracted from a set of Tools T (Figure 1, Left).

In order to formalize the problem, we must define an injective relation $\Re$ that assigns every triple (m_i, p_j, i_k) to one or more tools t_n belonging to the set of tools **T**. That can be formally expressed as:

$$\Re \subseteq X \times T,$$

$$\text{where } \mathbf{X} := \mathbf{M} \times \mathbf{P} \times \mathbf{I} = \{(\mathbf{m_i}, \mathbf{p_j}, \mathbf{i_k}): \mathbf{m_i} \in \mathbf{M}, \mathbf{p_j} \in \mathbf{P}, \mathbf{i_k} \in \mathbf{I} \}$$

Where the restriction **R** will map to a subset of the available tools **T** as shown in the following expression:

$$\mathbf{R_{(mi, pj, ik)}} = \mathbf{t_i} \cup \mathbf{t_j} \cup \dots \cup \mathbf{t_r}$$

The allowed tool set (**AT**) is then described by the following expression:

$$\mathbf{AT_{(mi, pj, ik)}} = \overline{(\mathbf{T} \cap \mathbf{R_{(mi, pj, ik)}})}$$

4. STEEL DETAILING CAD INTERFACE GENERATION

In this section, we present a sample application based in our approach introduced in chapter 3. Depending on the sub process (process stage), model and user intention we semantically generate a structural CAD User Interface containing the recommended tools. In order to generate the UI, we use process ontology whose classes and relations are modeled according to an international standard for the specific domain (CIS/2).

4.1 Modeling of the Process ontology according to CIS/2 standard

The structural steelwork lifecycle supported by CIS/2 is shown in Figure 2. This is a high level overview of the process model. CIS/2 describes three major stages needed to design a structural steel frame for a building: (i) Analysis, (ii) Design and (iii) Fabrication. Each phase has its own data model that serves for information exchange between each other and possible external applications. The three different models of a steel structure are called "Views" [5]. These views served as the initial source for our process ontology implementation where they are used as a simplified model of the structural design process (this process ontology will be extended in a future work to reflect a more explicit representation).
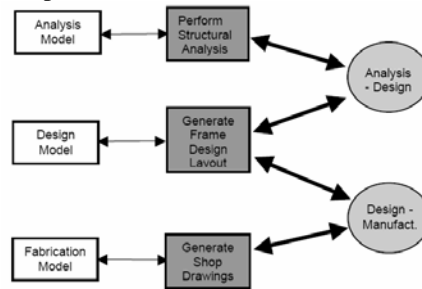

Figure 2 – Simplified process model identifying the various CIS/2 views and major exchanges, supported by high-level conformance classes [1]

4.2 Knowledge representation using Ontologies – The OWL approach

The existence of tools for ontology definition, querying and reasoning gives interesting possibilities to several application domains, including the engineering domain[12]. An ontology is the explicit specification of a conceptualization [6]. In simple terms, it is a description of the concepts and relations in a domain for the purpose of enabling knowledge sharing and reuse. A body of formally represented knowledge is based on a conceptualization: the objects, concepts and other entities that are assumed to exist in some area of interest and the relations that hold among them [6]. A widely used ontology language is OWL, which is a semantic mark-up language for publishing and sharing ontologies on the World Wide Web. OWL is developed as a vocabulary extension of RDF (the Resource Description Framework) and it is derived from the DAML+OIL Web Ontology Language allowing more expressiveness than RDF depending on the OWL flavour used [10]. There exist several OWL API's to handle computationally an ontology, between the most known ones, we can mention the Protégé OWL API [10], KAON2 [9] and the WonderWeb OWL API [11].

Our Process Ontology implementation is divided in two main classes: Available Semantic Tools and Tasks. The first contains the available tools and the second, the overall actions, as composition of a Model type (steel or wood), sub Process (analysis, design or fabrication) and User Intention (assembly, detailing or review). In order to simplify our implementation, our set of tools contains as a proof of concept three plug ins developed within the framework of a research project. Our test tools are: (i) a Virtual Reality (VR) visualizer that provides a 3D enhanced visualization of a large data set, (ii) a Dimensioning Tool (DIM) which dimensions special beams for workshop manufacturing and (iii) an Element Generation Tool (GEN) which generates structural CAD objects automatically. In figure 3, a detail of the ontology modeling (with some OWL restrictions) is shown.
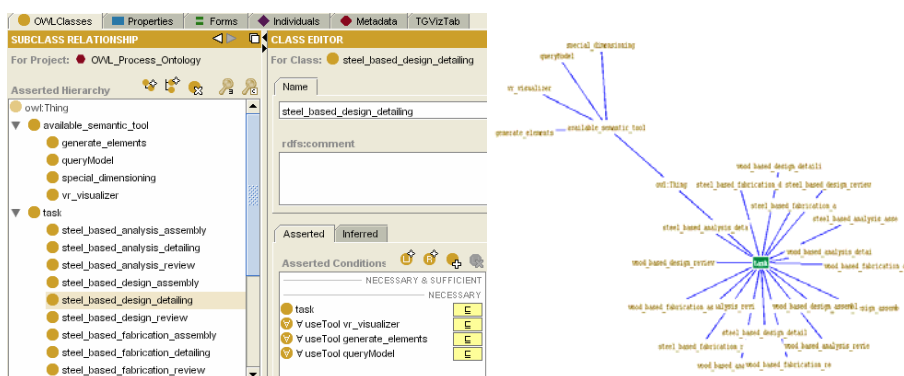

Figure 3 – Detail of the Process Ontology based on CIS/2

4.3 Modeling of the restrictions for tool selection in OWL

As stated before, a combination of model type, process stage and user intention produces a set of available tools. To implement $\Re$ (the injective relation) in the

ontology, the open world assumption and OWL description logics are used in order to relate both parts of the ontology via a restrictions. Description Logics are a family of knowledge representation languages which can be used to represent the terminological knowledge of an application domain in a structured and formally well-understood way. The ontology implementation of a logic for the description of such combination usually requires the use of a reasoner like Pellet or RACER [10].

We simplify the problem by creating a sub class of the Task class for each element in the combinatory set (figure 4). This means, for example, that for a modeled subclass inside the ontology (e.g.): *steel_based-analysis-detailing*, a table with relations can be constructed in where a simple map to the allowed tools can be modeled. For the example case (*steel_based-analysis-detailing)*, the allowed tools set will contain the Virtual Reality (VR) visualizer and the Element Generation Tools (GEN). In figure 4 part of the test relations table used in our example is presented.

model_type	process_stage	for	user_intention	useTool	VR	DIM	GEN
steel_based	design		assembly				
steel_based	fabrication		assembly				
steel_based	analysis		assembly				
steel_based	design		detailing				
steel_based	fabrication		detailing				
steel_based	analysis		detailing				
steel_based	design						

Figure 4 – Detail of the relations table

In order to model the mapping from the Available Tools (**AT**) subclass of the ontology and the task class, we restrict an object type property using OWL DL (description logics) restrictions. The restriction for this specific case is:

$$\textbf{R = Dimensioning Tool}$$

$$\textbf{T = Visualization Tool} \cup \textbf{Element Generation Tool} \cup \textbf{Dimensioning Tool}$$

$$\textbf{AT} = \overline{(\textbf{T} \cap \textbf{R})} = \textbf{Visualization Tool} \cup \textbf{Element Generation Tool}$$

Apart from the three tools to choose from, we have also implemented a fourth tool that will be presented in every set of allowing tools (and hence in every UI). This tool is called "Query Tool" and provides the user with means to semantically interrogate the geometries contained in the CAD model. We generate also a menu operations tool called "options" in order to perform maintenance operations like for example, destroy a semantically generated UI in the case that the user wants to create a new one.

For the implementation of the prototype, we used AutoCAD 2006 running Pro Steel V17, which is a commercial software that runs as an add-in to the base CAD extending the capabilities of the system from the structures design point of view. For the modeling of the CIS/2 ontology, we used Protégé interfacing with the ontology with the provided OWL API. In the Figure 5 a screenshot of a the test CAD with a generated semantic UI for the example is shown.

For testing the prototype, the presented schema was implemented as a part of a research project called MiroView where the ontology supported UI is being

used for the recommendation of CAD tools in an actual steel detailing company. At present time, tests are still in an ongoing stage therefore the results will presented in a future work.
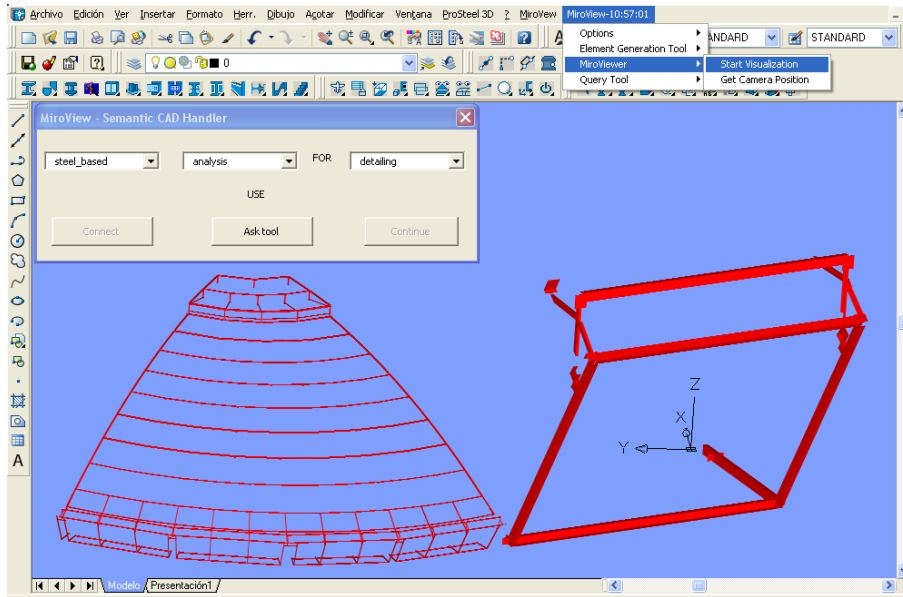

Figure 5 – A Semantically generated UI

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a Knowledge based approach for the exploitation of semantic aspects (e.g. user intentions and tasks) for the real time automatic generation of graphical user interfaces on a structural CAD software. This approach deals directly with the structural design process in order to suggest the user with the most appropriate tools for every stage of the design. The presented schema was modeled using an ontology that allows the semantic interrogation of the different parameters needed to produce the menu. As test tool, we presented an example based in the methodology with three test tools and a simplified process division. As future work we will extend the presented results in order to use more tools. We will extend in a future work the process ontology in order to have a more precise stage division, allowing a better tools recommendation. Also as future work, the different tests in a real environment will be analyzed in order to check the efficiency of the methodology.

5.1 Acknowledgments

to apply the techniques presented in this paper. We also thank SOME for their participation in this research project.

6. REFERENCES

1. CIS/2 – CIMSteel Integration Standard, WebPage: http://www.cis2.org/ Last access 26 January 2006.
2. Ekholm: "Activity objects in CAD-programs for building design". In: Proceedings of the Ninth International Conference on ComputerAided Architectural Design Futures Eindhoven, 2001, 61-74
3. Fridman, McGuinnes: "Ontology Development 101: A Guide to Creating Your First Ontology" Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880, Stanford Knowledge Systems Laboratory, Mar 2001.
4. Furtado E, Furtado J, Silva, et al: "An ontology-based method for universal design of user interfaces", Proceedings of Workshop on Multiple User Interfaces over the Internet: Engineering and Applications Trends, Lille, 2001.
5. Georgia Tech CIS/2 HomePage, WebPage: http://www.coa.gatech.edu/~aisc/ Last access 26 January 2006.
6. Gruber TR: "Toward principles for the design of ontologies used for knowledge sharing". International Journal of Human-Computer Studies. 1995. Volume 43 , Issue 5-6   Nov./Dec. 1995. Pages: 907 - 928. ISSN 1071-5819.
7. Hovestadt, Gramberg, et al. "Hyperbolic User Interfaces for Computer Aided Architectural Design", Conference on Human Factors in Computing Systems Denver, Colorado, United States , 1995, 304 – 305.
8. Igarashi. "Supportive Interfaces for Creative Visual Thinking" Collective Creativity Workshop, Nara (Japan), 2000.
9. KAON2, WebPage: http://kaon2.semanticweb.org/ Last access 26 January 2006.
10. Knublauch, Musen, Noy. Tutorial: Creating Semantic Web (OWL) Ontologies with Protégé. International Semantic Web Conference (ISWC2003).
11. WonderWeb OWL API HomePage, WebPage: http://owl.man.ac.uk/api.shtml/ Last access 26 January 2006.
12. Posada J, Toro CA, Wundrak S, Stork A: Ontology Supported Semantic simplification of Large Data Sets of industrial plant CAD models for design review visualization. In LNCS - Lecture Notes in Computer Science / Artificial Intelligence (Selected papers from KES05). Springer Verlag GmbH. 184-194.
13. Puerta, Eriksson, Gennari: "Model-Based Automated Generation of User Interfaces" International Conference on Intelligent User Interfaces. San Francisco, California, United States, 1997., 65 – 72.