INTERNATIONAL DESIGN CONFERENCE - DESIGN 2010
Dubrovnik - Croatia, May 17 - 20, 2010.

EMBEDDING DESIGN RATIONALE CAPTURING IN PLM SYSTEMS – A CASE STUDY WITH IBIS-BASED DIAGRAMS

N. Pavković, N. Bojčetić, I. Vadla and D. Rohde

Keywords: IBIS diagrams, design rationale, PLM

1. Introduction

Standard design documentation is a description of only the final design, without insights to "design process history": concepts, alternatives, decisions, etc. Design rationale (DR) can offer more: not only the decisions, but also the reasons behind each decision, including its justification, other alternatives considered, and argumentation leading to the decision [Lee 1997]. This information provides important insights into previous designs for other business processes within an enterprise, e.g. the redesign of a product, provision of service to customers, etc. [Wang 2009].

Despite over 30 years of research, there are still few rationale systems used in practice [Burge, Bracewell 2008]. There is a strong consensus that rationale is very valuable, but there is an equally strong concern that the costs of its capture may be too high. In order to justify the costs of its capture, it is essential to establish ways in which rationale can be useful that exceed the simple provision of additional design documentation [Burge, Bracewell 2008]. Design rationale capturing tools are beginning to be accepted in industry, e.g. the Design Rationale editor (DRed) developed by Engineering Design Centre (EDC) of Cambridge University [Bracewell et al. 2009]. Preliminary research on DRed usage in practice shows that structured information is easier to interpret than traditional design definition reports [Aurisicchio 2007].

Product Lifecycle Management (PLM) system offers efficient platform for document management and information management across the product lifecycle. We want to explore how these capabilities of PLM systems could be used to manage the process of design rationale capturing and retrieval in integrated product development teams. In other words, the aim of this research is to explore the methods and possibilities of embedding the process of design rationale capturing in commercial PLM system. We expect (and believe) that modelling, programming and customization tools and functionalities offered in PLM system are suitable and powerful enough to fulfil the development requirements for embedded design rationale capturing and retrieval tool. Especially, we want to explore further potential benefits of such approach (price and time of development, possibilities of dissemination, extension and/or customization of the proposed model).

Motivated by the success story of DRed tool, we have developed a prototype of "Issue based information system" [Kunz, Ritel 1970] (IBIS) software tool comprising elements similar to DRed. Our model has key IBIS elements, but is still far from full richness that is possible to be defined in a DRed model. The aim of this tool is to provide a graphical interface for creating and manipulating elements of captured design rationale in the environment of PLM system. Elements of IBIS diagram are stored as objects (class instances), and are treated in PLM system as any other class of document. This tool has been developed in a way that enables tight integration (embedding) with programming and customization capabilities of particular commercial PLM system. Such an approach has been chosen because our intention was to explore the possibilities of managing particular elements of design rationale (e.g. arguments, questions, answers etc.) using the usual mechanisms of PLM system. In other words, we want to go one level deeper from the level of managing design rationale document (in this case IBIS diagram) - to the level of managing elements (fragments) of design rationale documents. We believe that major benefits of this approach could be shown in usage of existing "standard" PLM team communication, storage, maintenance and especially search capabilities. Furthermore, we want to proceed with development of new design rationale management subsystem within a PLM framework. Proposed approach will include the development of new classes of documents and their fragments (elements) together with new procedures that will be embedded in "standard" PLM functions.

The main objectives are:

- to develop simple as possible, but still efficient enough system that will encourage designers to capture and reuse design information and knowledge;
- to identify and propose filter and query mechanisms embedded in PLM system for knowledge search and retrieval
- to offer the users (designers) an environment which is easily customizable for their particular contexts and domains

The scope of this research is the improvement of knowledge management process for embodiment and detail design tasks that are parts of complex and long-lasting product development projects which repeat in cycles of 3-5 years.

Aforementioned requirements could be associated (and divided according) to following issues:

- managing of team communication, workflow and permissions in the process of knowledge capturing, as well as in later processes of product lifecycle (manufacturing, servicing, etc.),
- semi-automatic classification of captured and stored knowledge elements (particularly their associations with other entities managed in PLM system) which will provide one of the searching methods for knowledge retrieval,
- searching of stored knowledge elements through sets of instances of their classes and attributes as well as through associated sets of instances of classes of projects, users, products etc.

This approach could also be viewed as extension of "standard" PLM system.

The research questions that will be considered in this paper are:

- could the proposed approach (of embedding design rationale management in PLM system) be the faster (more efficient) way to answer previously considered requirements comparing to development of an entirely new information management system
- is it possible and useful to implement predefined elements and structures for generating parts of design rationale explanations, e.g. like sets of "pro" and "con" arguments for narrow domains specific in particular design environments (the purpose of these sets would be to ease and speed up the capture and indexing processes)
- could such an approach encourage designers to capture and reuse more design information and knowledge than they would "normally" do in their usual design routine

We assume that with proposed approach we could develop information system and search mechanisms that will be able to answer following examples of knowledge retrieval requests:

- Did a similar design problem arise in some of the previous projects, and if, in which one? If such project(s) is found, how the design problem was solved?
- Designer wants to find out more about someone's experience with: one particular approach, one partial solution, etc.
- Which were the main problems in a set of similar previously finished design tasks and projects, what errors have been made?
- What were the reasons of selecting particular design solution (alternative) in previous stage/project, what were the reasons for rejecting other alternatives?

1.1 Related work

Huet et al. (2009) report a preliminary investigation into the management of digital sketches through PLM solutions. They propose a preliminary classification of sketches in order to facilitate their management and storage in PLM systems. Integration scenarios are also suggested using current PLM capabilities.

DRed is a graphical software tool for design rationale capture principally in the early stages of design and in resolving in-service problems. The first generation of DRed tool [Bracewell, Wallace 2003] has been focused on the capture and communication of "Design for X" knowledge such as manufacturing considerations and on problem diagnosis. Second generation [Bracewell et al. 2009] has been extended to capture and communicate behavioural and functional design knowledge. Its Issue Based Information System (IBIS) based design argumentation is integrated with hierarchical Functional Analysis Diagrams (FAD), a form of Concept Map.

McKay et al. (2009) have developed a prototype software tool that incorporate e-drawings, design rationale maps from a rationale editor (DRed) and allows users to visualize structures and relationships within and across product, process and enterprise network structures.

Wang et al. (2009) aim to develop methods and tools for designers to facilitate the re-use of design rationale captured via DRed. Their first step is to improve the key-word based search by providing better human-computer interaction. The improved retrieval is implemented by a novel combination of three methods: suggesting potential keywords to designers, quantifying the relevance of retrieved information and recommending relevant information based on the dependencies.

Aurrisicchio and Bracewell (2009) present research to extend the functionality and the notation of the DRed tool to support the generation of new diagram types. A novel approach to designing and its documentation by integrated diagrams is proposed, formalised into a templated structure and illustrated by means of a case study in the aerospace engineering industry. They promote the idea of using a set of diagrams to represent functional, behavioural and structural thinking in design.

Compendium [Buckingham Shum et al. 2006] is the semantic hypertext concept mapping tool at the heart of the Compendium methodology. It is the result of over 15 years' continual research, deployment and development of a tool to support the real time mapping of discussions in meetings, collaborative modelling, and the longer term management of this information as organizational memory.

Burge (2008) presents the results of a pilot survey of software developers who were asked how they would envision using rationale and what they believe the most important barriers are. Although some results were as expected, there were also some surprises.

2. Modelling and embedding of IBIS based diagrams in PLM system

Extensive usage of IBIS diagrams in designers’ everyday work shows a need for tracking or storing previous versions of IBIS elements (or objects) in order to reuse them in future. Storage can be achieved in different ways - for example saving the whole object’s state in different files, saving only changes, etc. Our approach is, as we like to think, a pragmatic one – to use the currently available technologies that designers are already familiar with. This is the reason for focusing our attention to the PLM applications and how to embed IBIS based diagrams in them. PLM applications offer many useful techniques and methods for information and documentation management. The techniques that are selected for the proposed approach are *document management* (Check in/Check out/Release) and *search algorithms and mechanisms*. Well known functionality of the document management mechanism is to enable the tracking of previous versions of working object. This functionality can be used to store and later to retrieve (trace) changes (or versions) of the IBIS diagram elements. It is obvious that showing every version of each element on the diagram would be inefficient and diagram will become unreadable. Therefore we choose to show only the latest versions of the elements on the IBIS diagram. Ability to define document as *final* or *release* is an important one because it shows to all the stakeholders that this thread or discussion is closed and solved. But this doesn’t mean that designer cannot reinitialize already closed thread. If this event should happened then new release of selected issue will be created and all new changes can be tracked and shown. In current state of the prototype implementation we decided to implement document management mechanism only at the level of the issue.

To demonstrate the proposed approach (embedding IBIS in PLM) we have developed following elements:

- An initial class structure of IBIS based diagrams and their elements that has been defined and added in PLM system, together with relations to "standard" basic PLM classes (fig. 1 and 2).
- Query examples for searching sets of IBIS diagrams, their elements and attributes stored in PLM system (figures 3 and 4).
- The prototype of IBIS based design rationale capturing tool embedded in the PLM programming environment. This prototype tool has two main functions:
  - graphical interface for creating and viewing IBIS diagrams (figure 5).
  - storage and maintenance of IBIS diagram elements as objects in PLM system (fig. 6).

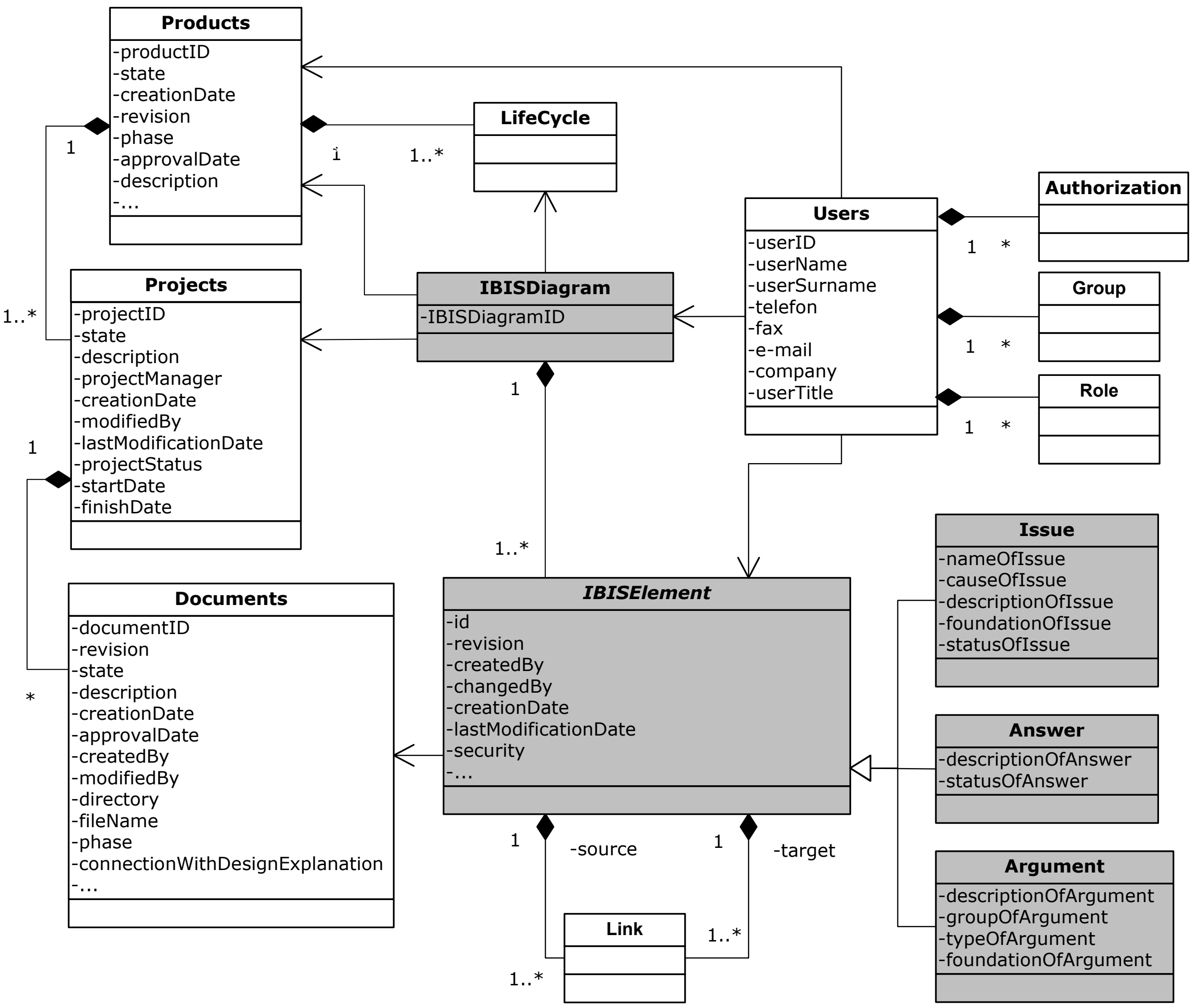

Figure 1. Proposed UML model of IBIS diagrams embedded in PLM system

Usual method of structuring information within PLM system is by means of association with product structure or project structure. Therefore Class "IBIS diagram" is associated with "standard" basic PLM classes "Projects", Products" and "Users". Class "IBIS element" is associated with PLM class "Documents" to enable linking with any kind of product documentation.

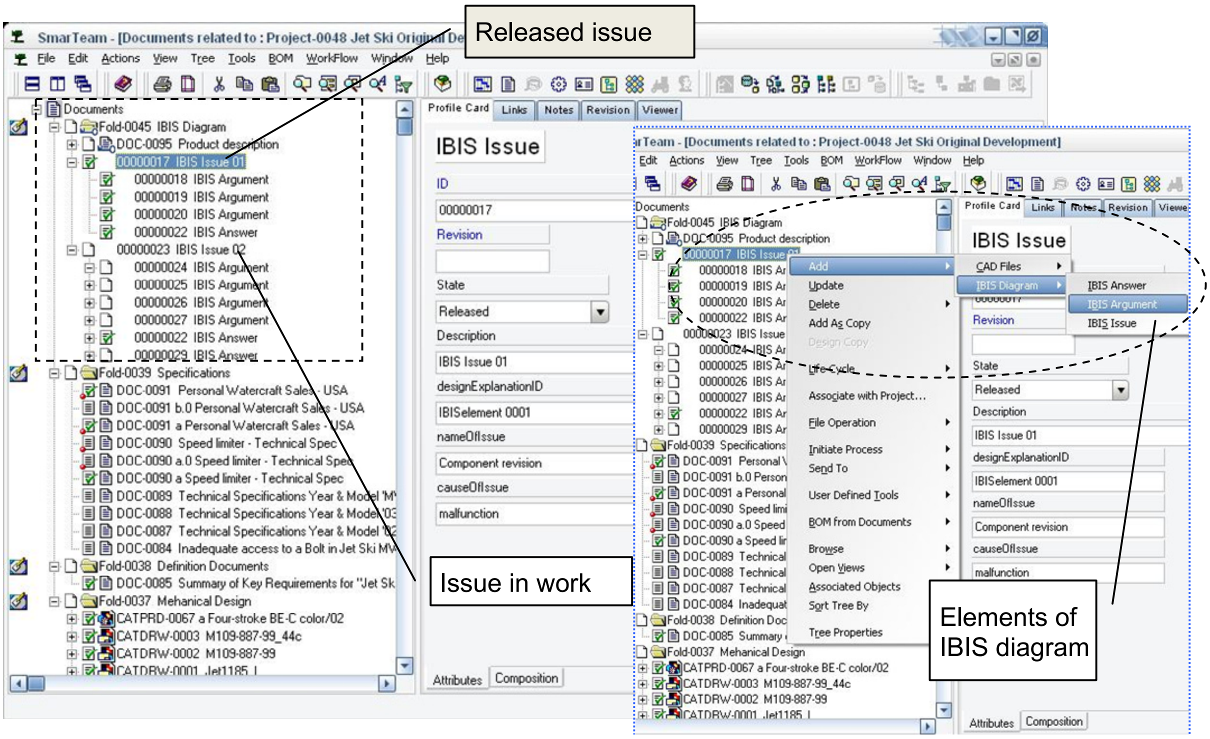

Figure 2. An example of IBIS diagrams and their elements as documents in PLM system

2.1 Benefits of the proposed approach in knowledge retrieval issues

We argue that one of the major benefits of the proposed approach is the ability to use PLM database functionality. In order to efficiently use PLM storage and search capabilities, the creation of "intelligent" links between IBIS elements are required. Here "intelligent" link refers to a link created as an object, not just as a pointer to another object. We propose creation of several classes of link objects. Each of them is for specific object to object relation like issue-to-argument, issue-to-answer, argument-to-argument, etc. but they all are specialization of general link class IBIS_element-to-IBIS_element. Using this type of link, we could use the built-in PLM search for:

- simple requests - as finding the objects with specific attribute value,
- more complex requests (queries) – to find objects with specific attribute or attributes' values that are linked to another object with his own attribute value that are of our interest

2.2 Query (search) examples

If user wants to find all issues associated with particular product (figure 3) he or she needs to add product class and issue class to the search (query) window and to select the type of link he or she is interested in. After all classes are added, user can select attributes from each class and define search criteria. In this way user is indirectly creating an SQL query that will be run against the database. If execution of the defined search finds the data that match the defined criteria, this data will be displayed in another window. From this new window report can be created and exported in different file types.

Another example of useful search is to find out where or in which issues particular argument is used. In this case user selects particular argument from the IBIS Diagram tree and executes "where used" method or the user can define another search by selecting issue and argument classes and "where used" link class (figure 4).

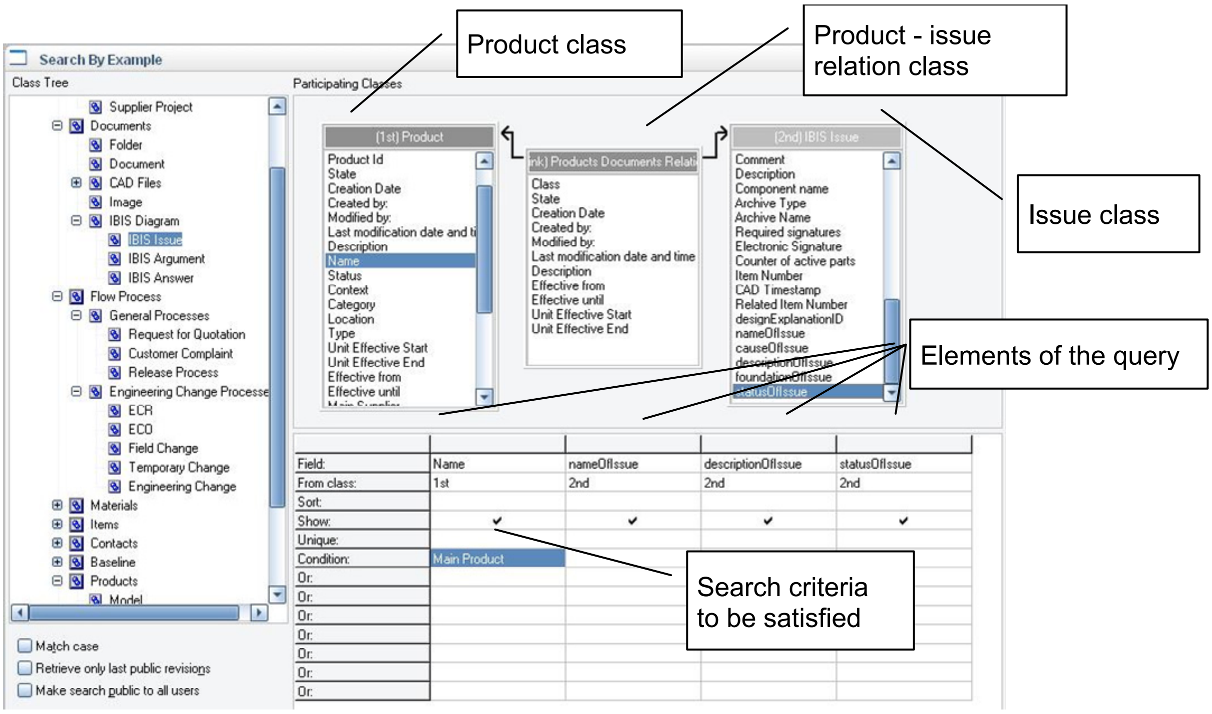

Figure 3. An example of searching through set of issues associated with particular product

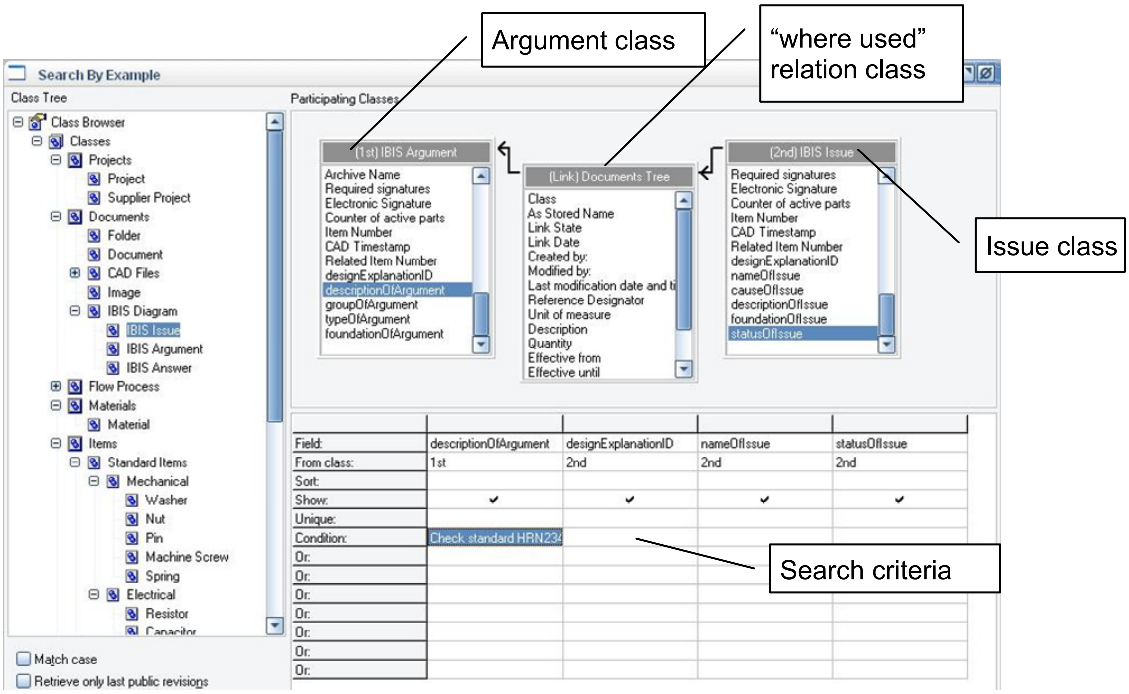

Figure 4. An example of using "where used" relation class

2.3 User roles and authorizations

In the development of the proposed approach it became obvious that IBIS model have to be tailored to be able to coexist with PDM application. Every IBIS element in our prototype application is considered as a different document object type (we were also considering the creation of one

document type with one attribute denoting IBIS element type, but we abandon this idea because of problems with link objects and the search strategies).
Security or Role/Authorizations functionality of the PLM system is still not mentioned in the detail, but it is also crucial for efficient implementation and embedding of the IBIS in the PLM system. Every user is authorized to read or read/write different objects. This is defined with roles and initial security level assigned to him. For the purpose of test implementation of the IBIS subsystem we propose the following roles:

Table 1. A proposal of user roles in manipulation of IBIS diagrams

	Possible operations with IBIS objects				
User role	Access to the IBIS element	Create new issue	Create other IBIS element	Use document management functionality	Propagate objects to the Release state
ibis_issue_creator	+	+	+	+	-
ibis_issue_moderator	+	+	+	+	+
ibis_participant	+	-	+	+	-
ibis_viewer	+	-	-	-	-

3. Graphical Interface development

This chapter gives an overview of the methods and techniques used in development of prototype software tool that is necessary to realise embedding and manipulation of IBIS diagrams in PLM system. Graphical interface is necessary to work with IBIS diagrams, therefore this was the first component developed in a way that it could be tightly integrated and embedded in programming and customization environment of PLM system [Vadla 2008 and 2009]. After solving graphical issues the second step was to add the functionality of storage and maintenance of IBIS diagram elements as objects in PLM system. This functionality is provided by using object-oriented programming language supported by particular PLM system.
Created graphical interface for design rationale capture is based on the well known IBIS system, where we use *issues (problems)*, *answers* and *arguments* as graph's key elements. These elements are predefined and can take on different predefined properties such as *solved*, *accepted, rejected, active, not active, pro and con*. The tool for IBIS graph creation was created using one of the mainstream programming languages form the Microsoft Visual Studio package, namely C Sharp (C#) language. C sharp is an object-oriented language which has its roots in C++ and Java programming languages. The form for manipulation with the graph's elements is the core part of the prototype tool. To leave as much "clear" space on form as possible, there are no buttons or commands on it. The elements are created and managed through context menus (figure 5) which are different depending on the element we select. The basic elements of the IBIS diagram are realized through several classes in C# programming language. Each element's visual appearance and properties are defined in separate classes, and are created and manipulated using additional classes. Using this structure, the IBIS elements remain independent, which is a result of an approach to manipulate each basic IBIS element separately inside the PLM system.
Two additional functionalities have been implemented in prototype tool:

- the ability to keep record of the design rationale creation flow - after each action on one of the graph elements, an event is recorded and stored in the "event history tree" (figure 5),
- the ability to mark some words within elements of the graph as keywords for later search and retrieval (also see figure 5).

Figure 5 shows a part of IBIS diagram example developed for international student design project. Aforementioned elements of IBIS diagram and graphical interface are indicated with textboxes on figure 5. The users select the desired actions from the context menus depending on the element that is being managed. These actions comprise creating IBIS diagram elements, as well as changing their properties.

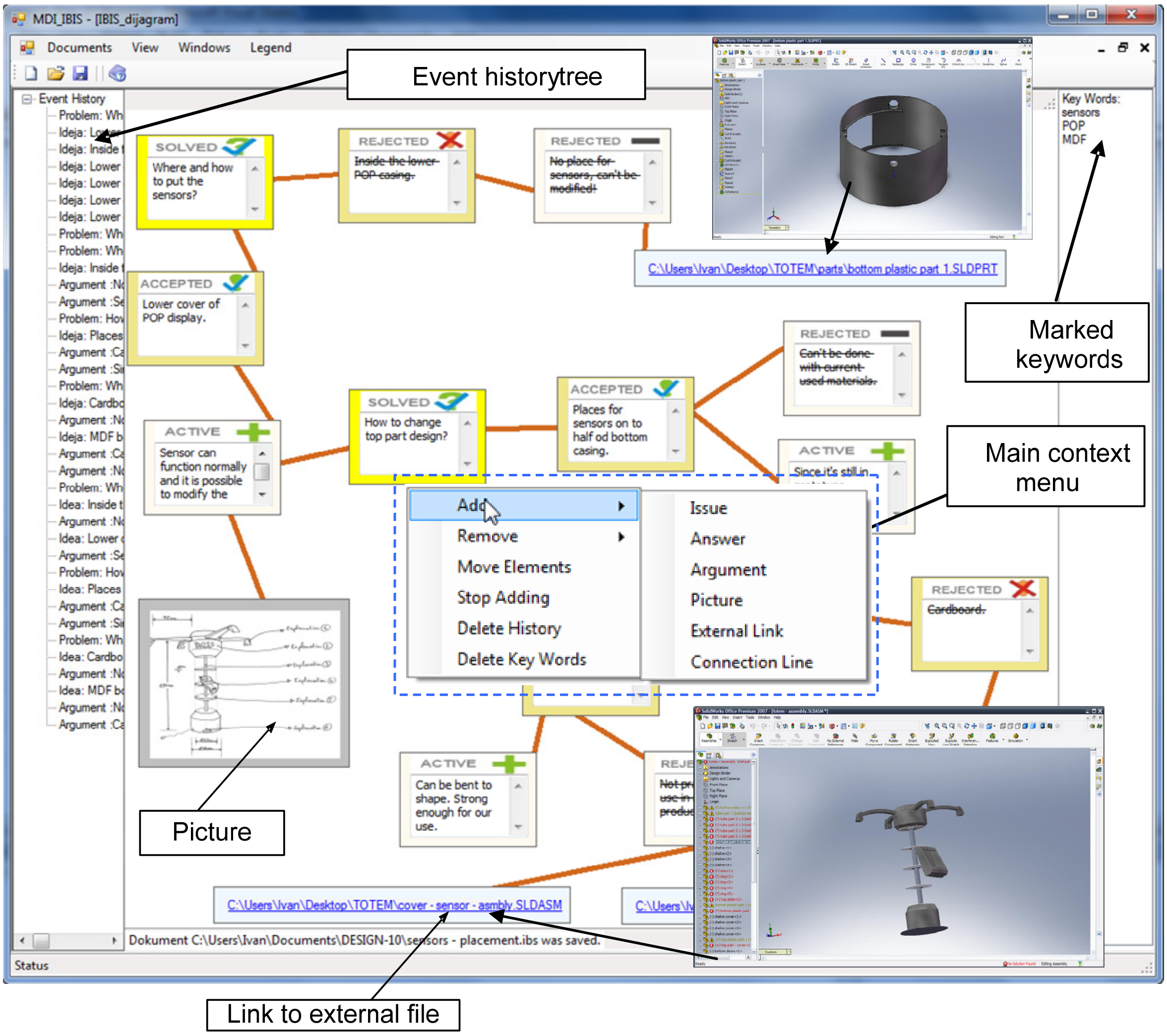

Figure 5. Graphical interface for creating and maintaining an IBIS diagram

4. Overall usage scenario

Usage scenario for proposed model and prototype tool implementation includes following steps:

1. Initial definition of all proposed classes and their attributes (as shown on UML diagram on figure 1). This step should be done only at the beginning of the implementation process in particular design environment.
2. Defining the new particular IBIS diagram as the new instance of "IBIS diagram" document subclass
3. "Opening" selected IBIS diagram with graphical interface tool, creating and manipulating diagram elements (figure 6)
4. Storing created diagram state as set of objects, their attributes and relations in PLM system. This function is provided with application integrated with graphical interface tool (figure 6).
5. Check in, check out, propagation, maintenance and search operations are being performed using "standard" PLM forms and tools (figures 2, 3, 4 and 6).

IBIS diagram elements could also be created with PLM system forms, but such option is surely not recommendable, because user is not able to view complete hierarchical structure of the diagram. Whenever the need for updating and upgrading for the particular diagram occurs, designers could start the graphical interface tool or in other words "open" the diagram. "Opening" the diagram means that application will read and show last saved state of all objects belonging to particular diagram. All objects' positions, relations and attributes will also be restored. Steps 3, 4 and 5 are being repeated in the life cycle of one particular diagram.

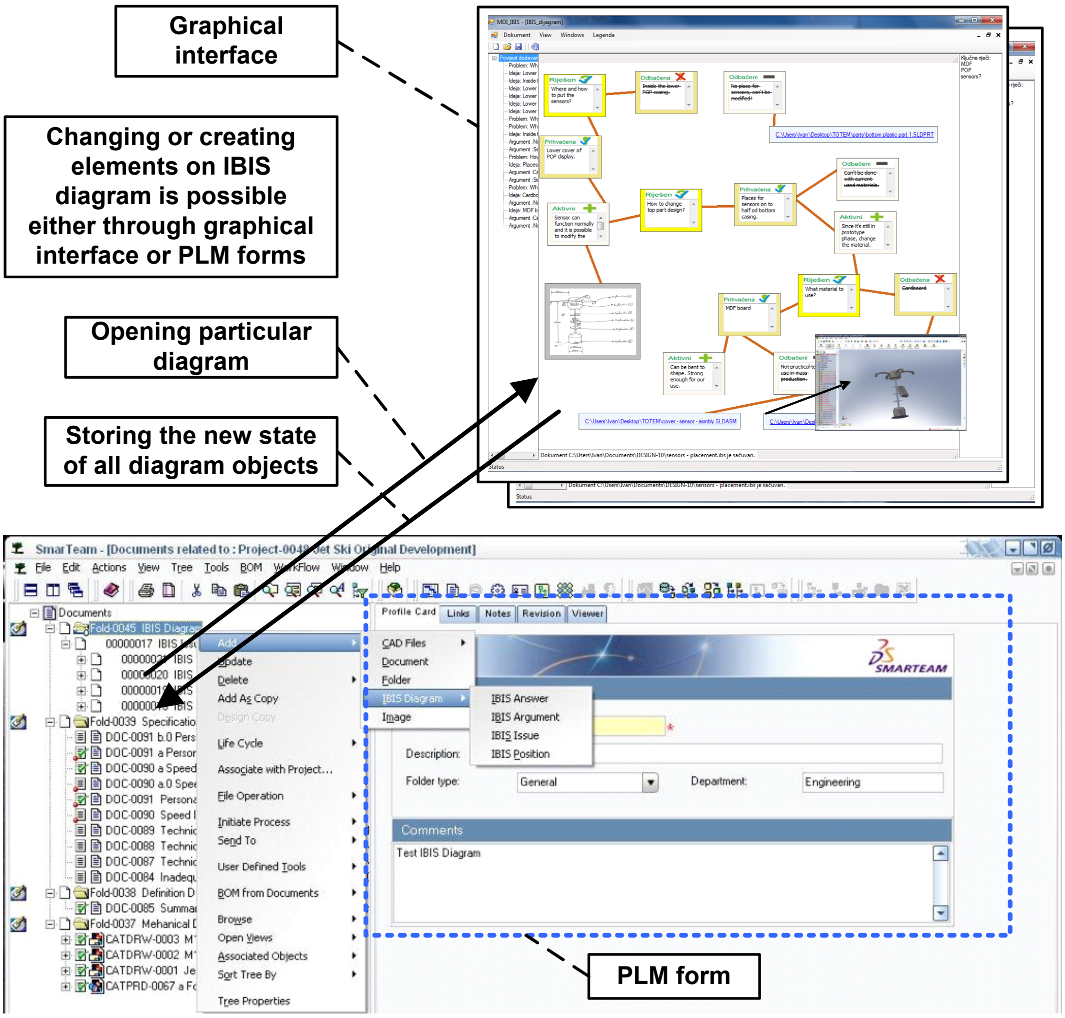

Figure 6. Two ways of creating and manipulating IBIS diagrams

5. Conclusions and future work

The goal of this research was to explore how the well known functionalities of PLM systems could be used to manage the process of design rationale capturing and retrieval in product development teams. We argue that such an approach could offer the following advantages:

- using of PLM system database and search mechanisms (especially "intelligent" relation objects) should provide efficient and simple procedures for retrieval of captured knowledge
- consistent and relatively well known methods and procedures for resolving team collaboration issues in simultaneous work and team sharing of IBIS diagrams
- using "standard" PLM mechanism for creation and maintenance of internal and external links to all kinds of product documentation
- possibilities to develop sets of "predefined" elements and/or queries that could make design rationale capturing and retrieval processes easier and faster

Of course, mentioned "advantages" must be measured and validated by experimental usage of proposed system in design practice. In parallel with practical validation, future work will be focused to some unresolved issues and dilemmas that could be sources of problems in practical usage, for example: how deep is necessary to use "standard" PLM document management mechanisms – the

compromise must be found between potential benefits and the amount of extra designers' work. As mentioned before, currently we decide to implement document management mechanism only at the level of the issue.
The major disadvantage of proposed approach lies in fact that the main parts of the prototype software tool must be developed and customized according to particular PLM system and his programming environment. However, we hope that presented approach could contribute to research efforts in design rationale capturing tools as well as in development of new generation of PLM systems that should be extended to broader contexts and domains outside the standard design documentation which serve as a description of only the final design.

Acknowledgments

This research is part of funded project “Models and methods of knowledge management in product development” supported by the Ministry of Science and Technology of the Republic of Croatia.

References

Burge J. E., Bracewell R., "Guest editorial of special issue: Design Rationale", Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2008), 22, 309–310.

Kunz, W., Rittel, H. (1970), Issues as Elements of Information Systems, Working Paper 131. Berkeley, CA: University of California, Berkeley, Center for Urban and Regional Development.

Lee, J. (1997). Design rationale systems: understanding the issues. IEEE Expert: Intelligent Systems and Their Applications 12(3), 78–85

Bracewell R., Wallace K., A tool for capturing design rationale, International Conference on Engineering design - ICED '03, Stockholm, August 2003, pp. 185-186

Vadla I. (2008), Graphical interface for IBIS-based design rationale capturing tool, Bachelor thesis, Faculty of mechanical engineering & naval architecture, University of Zagreb, (in Croatian)

Vadla I. (2009), IBIS-based Design rationale capturing embedded in PLM system, Master thesis, Faculty of mechanical engineering & naval architecture, University of Zagreb, (in Croatian)

Wang H., Johnson A, Bracewell R. (2009), Supporting Design Rationale Retrieval for Design Knowledge Re-Use, Proceedings of the 17th Int. Conf. on Eng. Design – ICED 09, Stanford University, Stanford, USA, 2009.

McKay A., Kundu S., Dawson P. G., de Pennington A. (2009), An Integrated Product, Process and Rationale Model for the Provision of Through-Life Information in Product Service Systems, Proceedings of the 17th International Conference on Engineering Design – ICED 09, Stanford University, Stanford, USA, 2009.

Huet G., McAlpine H., Camarero R., Culley S. J., Leblanc T., Fortin C. (2009), The Management of Digital Sketches Through PLM Solutions, Proceedings of ICED 09, Stanford University, Stanford, USA, 2009.

Bracewell R., Gourtovaia M., Moss M., Knott D., Wallace K., Clarkson J. P. (2009), Dred 2.0: A Method and Tool for Capture and Communication of Design Knowledge Deliberated in the Creation of Technical Products, Proceedings of the 17th Int. Conf. on Engineering Design – ICED 09, Stanford University, Stanford, USA, 2009.

Aurisicchio M., Bracewell R., (2009), Engineering Design By Integrated Diagrams, Proceedings of the 17th International Conference on Engineering Design – ICED 09, Stanford University, Stanford, USA, 2009.

Aurisicchio M., Gourtovaia M., Bracewell R. H. and Wallace K. M. (2007), Evaluation of how DRed design rationale is interpreted, Proceedings of the 16th Int. Conf. on Eng. Design – ICED 07, Paris, August 2007.

Burge J. E. (2008), Design rationale: Researching under uncertainty, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 22, 311–324.

Buckingham Shum, SJ, Selvin, AM, Sierhuis, M, Conklin, J, Haley, CB and Nuseibeh, B: (2006), Hypermedia Support for Argumentation-Based Rationale: 15 Years on from gIBIS and QOC, in, A Dutoit, R McCall, I Mistrik, and B Paech (eds.), Rationale Management in Software Engineering, Springer-Verlag, Berlin, pp. 111-132. [PrePrint:http://kmi.open.ac.uk/publications/pdf/KMI-05-18.pdf]

Neven Pavković
University of Zagreb, Faculty of Mechanical Engineering & Naval Architecture
Ivana Lucica 5, 10000 Zagreb, CROATIA
Telephone: +385 1 6168 545
Telefax: +385 1 6168 284
Email: neven.pavkovic@fsb.hr,
URL: http://www.cadlab.fsb.hr