Resolving Collaborative Design Conflicts Through an Ontology-based Approach

Moisés Dutra [a], Parisa Ghodous [b,1] and Ricardo Gonçalves [c]

[a] PhD Student, LIRIS Laboratory, University of Lyon 1, France, moises.limadutra@liris.cnrs.fr.
[b] Head of Collaborative Modelling Team of LIRIS Laboratory, Univ. of Lyon 1, France, ghodous@liris.cnrs.fr.
[c] Head of the Group for the Research in Interoperability of Systems (GRIS), Uninova Institute, Portugal, rg@uninova.pt.

Abstract. This paper presents an ontology-based approach to resolve conflicts in collaborative design. In a collaborative design environment, achieving a global design of a product implies the proposed model is realisable and acceptable to all participants involved in the design project. Whenever this not happens we have a conflicting situation. The work presented here is based on the use of ontology modelling (OWL) to represent knowledge and, like that, to enable a reasoning process to be done. The results of this reasoning, the conflicting axioms detected, are used as starting point to a conflict resolution process. First, an automatic approach is tried. In case of failure, the next step is the direct interaction among the project participants, i.e., negotiation and mediation. A small electrical connector was taken as example to illustrate our approach.

Keywords. Collaborative design, conflicts, ontologies, constraints, negotiation, case-based reasoning

1 Introduction

Time and resources required to resolve conflicting situations in collaborative design have increased proportionally to the complexity of modern industrial systems. According to [14], even more, companies use geographically distributed knowledge, resources and equipment. The collaborative design process is typically expensive and time-consuming because strong interdependencies between design decisions make it difficult to converge on a single design that satisfies these dependencies and is acceptable to all participants [7]. Concurrent engineering brings new ways of organising design and manufacturing activities, introducing

[1] Laboratory of Computer Graphics, Images and Information Systems (LIRIS); Bâtiment Nautibus, 43, bd. du 11 Novembre. 1918, 69622 Villeurbanne cedex, France; Tel: +33 (0) 4 72 44 58 84; Fax: +33 (0) 4 72 43 13 12.

deep modifications, such as the concurrent realisation of product life cycle tasks. The collaborative approach also emphasizes the integration of all disciplines that contribute to the product development. The early-stage design is a very important part of this approach, as important decisions are made considering the entire project life cycle [6, 12].

Hence, conflict attenuation and resolution in early-stage design are essential points to be considered. Conflicts can be extremely resource hungry in terms of resources such as development time, budget and materials. Preventing them at this point – rather than later – is preferable, as it enhances the chances of success for consecutive design phases. This process involves identification and categorisation of conflicts and notification to the different involved parts, in order to put the situation under control as soon as possible [10]. When early conflict detecting is not possible, or not successful, a conflict resolution process must be undertaken.

This paper presents an approach for conflict resolution in collaborative design that takes into account the results obtained by an ontology-based conflict detection process [2, 3].

2 Conflict dealing in collaborative design

A lot of approaches have arisen to deal with conflicts in collaborative design. Among them, we chose to highlight the following ones: ontologies; thesaurus; prototyping; constraints checking; constraints relaxation; case-based reasoning; rule-based reasoning; priorities management; negotiation and mediation.

Ontologies and thesaurus are resources used to resolve linguistic conflicts. While the use of ontologies permits dealing with more complex conflicts; providing exact terminology is an accurate approach to mitigate meaning-based conflicts – the polysemic ones. So, for this kind of conflict a thesaurus is suitable [4].

Simulation tools are used to detect conflict inconsistencies [13]. Virtual prototypes permit the detection of structural-level interferences and simulators permit the evaluation of objects being used in the design. The use of these tools envisages detecting eventual conflicts [12].

Constraints are used to represent system's requirements, in order to enhance the collaboration process. Requirements are represented as groups of variables in spaces of feasible values. Such spaces improve efficiency through avoiding artificial conflicts, improving design flexibility, enhancing change management and assisting conflict resolution [9]. A constraint checking is an automatic task, taken to verify the consistency of a given model. Defined constraints may be relaxed during the negotiation process – if it is necessary – to facilitate the search for a solution.

Case-based reasoning is the process of solving new problems based on solutions for similar past problems. In this case, the most common past solutions are taken as starting point to solve the new problem [6].

Rule-based reasoning takes predefined rules / statements as parameters to check the given model. It is quite similar to constraint checking, except that the rules defined by it are not design specifications but instead, they should be seen more like a to-do list, to be followed whenever a conflict appears.

The negotiation process involves direct interaction among designers, to find the best solution for everybody involved in the design project [8]. Priorities management and mediation are two tasks accomplished by the project manager, to solve the problem. Priorities can be attributed to designers, to knowledge areas, to specific topics, to product subparts, etc. They are used to establish a "rank of importance".

Mediation is a unilateral decision, made when the negotiation process fails and there is no more chances of success. It is an extreme solution and should be avoided as much as possible.

2.1 Using ontology modelling to detect conflicts

The use of ontology modelling in collaborative design has been proving to be a prominent approach to detect conflicts in early-stage design [2]. Besides, representing knowledge in Web Ontology Language (OWL)[2] offers a reasonable trade-off between expressibility and decidability, witch when used to verify product specifications in collaborative design may fit as an efficient conflict attenuator. OWL supports automated reasoning and, to this effect, has a formal semantics based on Description Logics (DL) – typically a decidable subset of First Order Logic; suitable for representing structured information about concepts, concept hierarchies and relationships between concepts.

The decidability of the logic ensures DL reasoners can be built to check OWL ontologies consistency, i.e., verify whether there are any logical contradictions in two or more ontology axioms. Furthermore, reasoners can be used to infer from the asserted information, e.g., to infer whether a particular ontology concept is a subconcept of another, or whether a particular individual in a given ontology belongs to a specific class. According to [11], a typical OWL reasoner provides at least the standard set of Description Logic inference services, namely: consistency checking; concept satisfiability; classification and realisation.

3 An ontology-based detection of conflicts

In our collaborative architecture [6], designers are grouped by clusters of knowledge and expertise. Each one of these clusters is called an agency. Inside the agencies, designers collaborate to achieve a common design model and ontologies are used to represent such models [3]. At this stage, intra-agency collaboration is done. Once this step is completed, common ontologies obtained there are merged together in a higher level, the inter-agency collaboration one. In this higher level, the common ontology obtained is, then, the final design solution (Figure 1).

[2] http://www.w3.org/TR/owl-features/

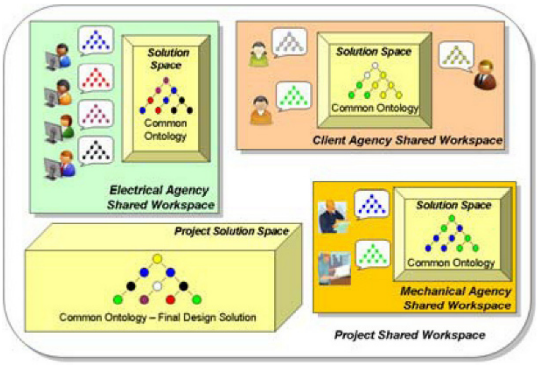

Solution Space
Common Ontology
Electrical Agency Shared Workspace
Solution Space
Common Ontology
Client Agency Shared Workspace
Project Solution Space
Common Ontology – Final Design Solution
Solution Space
Common Ontology
Mechanical Agency Shared Workspace
Project Shared Workspace

Figure 1. Collaboration levels and common ontologies

Each expert designs his own model according to his expertise and knowledge skills. A mechanical engineer will, naturally, be concerned about the mechanical structure of the product and his components. In the meantime, a thermal engineer will be more concerned about heating and temperature control for a specific part of the same product.

In this approach, both models – mechanical and thermal – are merged to a common one, comprising mechanical and thermal specifications. If we consider other knowledge – or interest – areas in such architecture, there will be one agency for each considered area. Thus, electrical engineers, material engineers, raw material suppliers, manufacturers, people of distribution department, clients, vendors, marketing people – among others, that is, every expertise involved in the design process, will be contemplated with an agency.

In our architecture is mandatory considering the publication of propositions in public spaces. Publishing a proposition means to merge different proposed instances of a product model into the design space. This merging is only possible if there is no interference between the elements, which means, if they are all coherent. To guarantee such a scenario, two operations are processed in the moment of the publication: constraint checking and ontology reasoning (Figure 2).

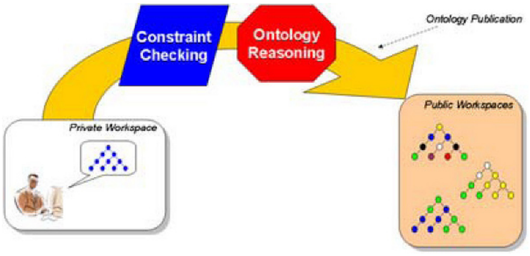


Figure 2. Ontology publication in public spaces

Constraint checking is part of the conflict attenuation approach. It uses predefined rules / statements to ensure the coherent data will be published. This step is not collaborative, as it does not take into account other proposals, but the one being published. However, as constraint checking is not enough to guarantee data coherence, an OWL reasoning is taken right after it.

To illustrate this situation, let's take an electrical connector as example. Such a connector comprises different subparts, e.g.: spring, shell, screw, cable, conductor, etc. Considering a small collaborative design project, where two engineers model the same piece differently, according to their personal convictions, two concurrent ontology instances will be given, representing the same product (Figure 3).

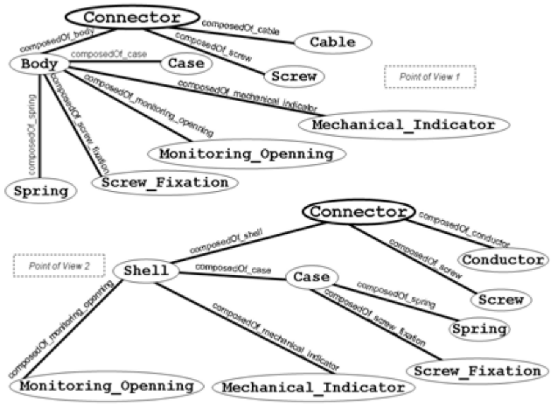


Figure 3. Two different points of view through the same product

As can be seen in Figure 3, second designer's *Shell* concept is equivalent to *Body* concept in the first designer's model. In the second one, however, *Spring* concept is no more linked directly to *Body* / *Shell* concept but instead, directly to *Case* concept. The conflict detection process (which comprises an ontology reasoning) undertaken in [2] has attested the inconsistency of these models, as well as the impossibility of merging them. In this case, we say *Connector* concept has been detected to be <u>unsatisfiable</u>. Discovering such information is essential to advance to the next step, the conflict resolution process.

4 The conflict resolution approach

In our architecture, all system data and information are stored in a blackboard (Figure 4). This blackboard comprises two subspaces:

- Solution space: This space stores the system database; merged ontologies (produced after the collaboration process); and predefined ontologies (if it is the case), to be used as "standard models" by designers.
- Collaboration space: Space where collaboration is done.

- Questions Area: Can be seen as a "FAQ" area of the system. It is used by designers to clarify global doubts / problems related to design process.
- Coordination Area: Project manager's workspace.
- Interaction Area: Is the designers' communication area. In there, they are able to notify one another, to leave them messages, to make them propositions, to make suggestions and to argue.
- Conflicts Area: This area is activated whenever a conflict is detected. It is the responsible for the conflict resolution process.

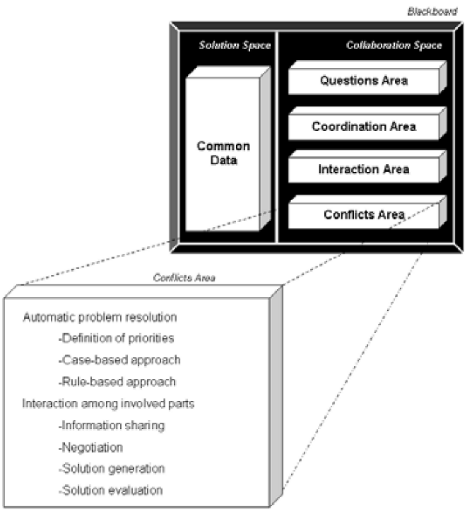


Figure 4. Blackboard and conflicts area

4.1 Resolving a conflict

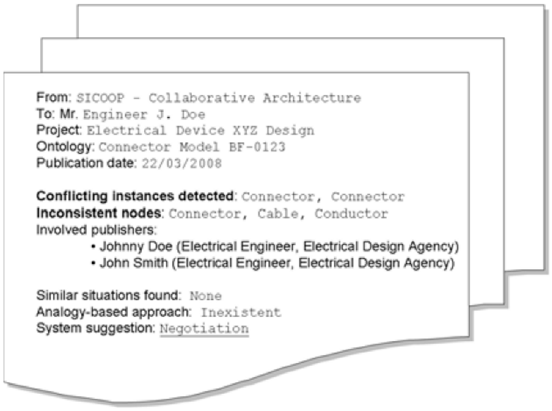
Once a conflict is detected the system tries, firstly, to resolve it in an automated way. To use such an approach, three options are available: definition of priorities, case-based reasoning and rule-based reasoning.

Definition of priorities is a project manager's task. He is the one in charge to state a specific designer / model has priority over another one. In this case, and if there is consistency, the higher ranked ontology will be set as the common model. Nevertheless, it must be highlighted that it is not mandatory defining priorities.

Next, the system will verify the past cases. Here it will use an analogy-based approach to make a decision, based on what have already happened before in such situations. However, as this step is not that easy because of the great number of involved variables (at the end, we can even say only rigorously identical ontologies can be compared), we do not take the achieved solution to be "the one" but rather, we send it to evaluation of concerned designers.

If some rules have been defined, a deduction process may take place. This step uses the same principles constraint checker does for conflict attenuation. Both approaches are based on ontology rules, expressed in SWRL[3] language. Here a constraint relaxation process may also be undertaken, if the project manager decides to.

If the conflicting situation persists at the end of the automated process, a negotiation process will be started. This time designers will directly take part of all phases of the task. First of all, each one of them – to whom conflicting situation matters – will receive a system notification saying an ontology merging has failed and a conflicting situation came up. This notifying message must "translate" to a comprehensible language the detected inconsistency, since designers are not necessarily ontology experts. In the example showed in Figure 3, *Connector* concept was detected to be unsatisfiable. Back there, incoherence was detected because both *Cable* and *Conductor* concepts – being disjoint – have been assigned as equivalent ones. Consequently, a typical notification like the one showed in Figure 5 is sent to concerned designers.


From: SICOOP - Collaborative Architecture
To: Mr. Engineer J. Doe
Project: Electrical Device XYZ Design
Ontology: Connector Model BF-0123
Publication date: 22/03/2008

Conflicting instances detected: Connector, Connector
Inconsistent nodes: Connector, Cable, Conductor
Involved publishers:
- Johnny Doe (Electrical Engineer, Electrical Design Agency)
- John Smith (Electrical Engineer, Electrical Design Agency)

Similar situations found: None
Analogy-based approach: Inexistent
System suggestion: Negotiation

Figure 5. System notification of the conflict

The recipient designer must, then, contact his partner(s) in order to resolve the situation. They should talk, attempting to produce together a feasible solution for everybody. They should also check each other's ontologies – by accessing the public workspaces (intra or inter agencies) – as well as they should ask for other designers' opinions, too. Finally, they are supposed to collaborate as much as they can to resolve the conflict. However, if achieving such a scenario is not possible by any reason, whatever it may be, a mediation process will be undertaken by the project manager. He is, in the last stage, the "referee of the quarrel".

[3] http://www.w3.org/Submission/SWRL/

5 Closing Remarks

Resolving collaborative design conflicts is a hard task to deal with. The differences among designers have to be taken account of, especially if we consider a very-large-scale design project. Each one of them has a different background, different expertise and different cultural and social points of view. It is never easy to resolve problems when such a set of people is involved.

In our collaborative architecture, we chose to use ontologies to model these different kinds of knowledge and expertise. We consider the use of OWL as a very efficient approach to represent knowledge in collaborative environments. Besides, OWL reasoning facilitates coping with conflicts in early-stage design, as it permits the detection of inconsistencies. The last trends in this research domain – along with the several ongoing projects that work with ontology merging and aligning [1] – have encouraged us to keep going toward this direction. Our latest results have been proving our expectations in this area, so far.

In our proposal, all resolution process relies on the OWL representation of information. Consequently, two different scenarios arise. The first one is the scenario where a standard ontology is given, i.e., defined before the starting of the collaboration process. In this context, designers should take this "standard ontology" and use it to build their personal models.

The second scenario is the one where we start from different kinds of representation. Here, each designer works with the format he is used to, e.g.: STEP protocols (ISO 10303), Function-Behaviour-Structure framework [5], natural language, among others. In this case, the collaborative platform is responsible for harmonising them, merging them into a common ontology. This common ontology is, then, used as "standard" for the reasoning process. The harmonisation process is the next step of our work.

6 References

[1] De Bruijn J, Martin-Recuerda F, Manov D, Ehrig M. State-of-the-art survey on Ontology Merging and Aligning. EU-IST Integrated Project (IP) IST-2003-506826 SEKT, Deliverable D4.2.1 (WP4), Digital Enterprise Research Institute, University of Innsbruck, 2004.

[2] Dutra M, Ferreira da Silva C, Ghodous P, Gonçalves R. Using an Inference Engine to Detect Conflicts in Collaborative Design. To appear in the 14th International Conference on Concurrent Enterprising (ICE 2008) – Lisbon, Portugal, June 2008.

[3] Dutra M, Ghodous P. A Reasoning Approach for Conflict Dealing in Collaborative Design. In proceedings of the 14th International Conference in Concurrent Engineering (CE2007) – Springer Verlag. São José dos Campos, Brazil, July 2007.

[4] Falquet G, Jiang, CLM. Conflict Resolution in the Collaborative Design of Terminological Knowledge Bases. In proceedings of 12th International Conference on Knowledge Engineering and Knowledge Management (EKAW), Juan-les-Pins, France, October 2000:156-171.

[5] Gero JS, Kannengiesser U. The situated function-behaviour-structure framework. In Design Studies, Elsevier, UK, 2004; 25; n. 4.

[6] Ghodous P. Modèles et Architectures pour l'Ingénierie Coopérative [in french]. Habilitation Thesis, University of Lyon 1, Lyon, France, 2002.

[7] Klein M. The Dynamics of Collaborative Design: Insights From Complex Systems and Negotiation Research. In Complex Engineered Systems, ISBN 978-3-540-32831-5, Springer Berlin / Heidelberg 2006:158-174.

[8] Klein M. Supporting Conflict Resolution in Cooperative Design Systems. In IEEE Transactions on Systems, Man and Cybernetics, Special Issue on Distributed Artificial Intelligence, 1991; 34, n. 6.

[9] Lottaz C, Smith IFC, Robert-Nicoud Y, Falting BV. Constraint-based support for negotiation in collaborative design. Artificial Intelligence in Engineering 14, Elsevier Science Ltd., 2000.

[10] Matta N, Corby O. Conflict Management in Concurrent Engineering: Modelling Guides. Proceedings of the European Conference in Artificial Intelligence, Workshop on Conflict Management, Budapest, Hungary, 1996.

[11] Sirin E, Parsia B, Cuenca Grau B, Kalyanpur A, Katz Y. Pellet: A practical OWL-DL reasoner, Journal of Web Semantics: Science, Services and Agents on the World Wide Web. In E. Wallace (Ed.) Software Engineering and the Semantic Web, 2007;5(2):51-53.

[12] Slimani K, Ferreira da Silva C, Médini L, Ghodous P. Conflict mitigation in collaborative design. In International Journal of Production Research, ISSN 0020-7543, Taylor & Francis, 2006.

[13] Sriram RD. Distributed and Integrated Collaborative Engineering Design. Savren, ISBN 0-9725064-0-3, 2002.

[14] Xie H, Neelamkavil J, Wang L, Shen W and Pardasani A. Collaborative conceptual design - state of the art and future trends. Proceedings of Computer-Aided Design, 2002; 34: 981-996.