INTERNATIONAL DESIGN CONFERENCE - DESIGN 2004
Dubrovnik, May 18 - 21, 2004.

QUESTION-DRIVEN MODELING

U. Sellgren

Keywords: design problem, engineering question, model requirements, modeling, scenario workflow

1. Introduction

Technology progresses by solving problems, that is, through reducing complexity by simplifying what was intractable and giving structure to what was ineffable. A problem is a state of difficulty that can be formally defined as the difference between a context-dependent actual state and a desired state. In engineering design, complex physical behavior and contradictory requirements, such as performance and reliability, frequently have to be addressed in a competitive environment where speed and agility are essential. Furthermore, many engineering problems are open-ended and complex. Engineering design can, thus, be viewed as a highly constrained and complex cognitive problem-solving process. Many engineering problems can be reformulated as questions with the form "What if …?". The aim of a problem-solving process is to find an answer to a question. When we find an answer to an engineering question, we create new knowledge or expand our knowledge, both of which are fundamental activities in engineering design [Hatchuel & Weil 2003], which is largely a knowledge-centered activity [Tomiyama 2003]. The design process makes use of different classes of knowledge. The acquisition, structuring, retrieval, and management of large amounts of knowledge is crucial to support design decisions [Ullmann & D'Ambrosio 1995].
There are only limited options for describing, acquiring, maintaining, administering, and getting rid of knowledge when it becomes outdated or obsolete [Vajna 2002]. Troxler [2002] defines knowledge as information sufficiently interpreted to enable action, where information is defined as raw data that has been structured and represented for human senses. In some cases, required knowledge is available somewhere in the organization, either in codified form [Boisot 1995] or in people's heads, or it may have to be sought outside the organization (e.g., on the Internet). In other cases, new knowledge has to be created and integrated with other types of knowledge, or available knowledge has to be expanded and deepened.

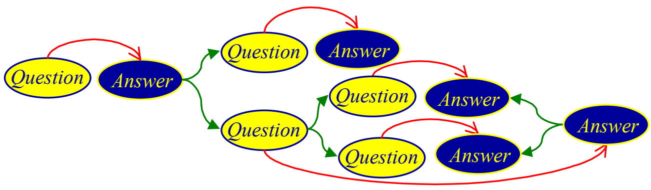


Figure 1. A cascading sequence of questions seeking answers

Design has the characteristic property of recursivity [Hatchuel & Weil 2003]. Design reasoning, which is a human cognitive activity, transforms problems into solutions, which represent new knowledge. However, it also transforms problems into problems, meaning that an answer to one question often triggers a new question. The search for an answer to a question sometimes requires a decomposition of the question into subquestions, as indicated in figure 1. The answer sought must be synthesized from the answers to the subquestions.

Models are important tools in complex cognitive activities such as design reasoning. A model is a simplified representation of some physical object or of an idea. Models can be classified as mental, physical, or analytical/numerical. A computer-based model is a cognitive construct that may be viewed as a container of codified information. In other words, it is basically a complex knowledge object. From this point on, the term 'model' will be taken as referring only to computer-based numerical models.

Simulation is the act of creating information by experimenting with a model. Fundamental to successful modeling is the ability to structure and simplify information. Thus, new information and new knowledge is also created in the modeling process itself, and not only in the use phase of the model. Modeling is consequently a cognitive activity that involves both analysis and synthesis. New computer technologies and new software for computer-based modeling and simulation open the way to new approaches to exploring and solving design problems and to expanding technological knowledge.

Systematic use of models to assist design reasoning (i.e., creating models that can be asked questions) offers significant competitive advantages, but it also introduces new types of complexity to engineering [Sellgren 2003]. An approach to engineering that significantly benefits from systematic use of models to generate, explore, enhance, and refine engineering knowledge may be referred to as a model-based approach. To achieve optimal performance, a model must be targeted at and designed to answer a specific question. Sometimes, a question can only be addressed by designing a completely new model, but in many cases it is possible to reconfigure a previously defined model, thus creating a new model variant. Reusing submodels in a new configuration, with a different context and purpose than the one for which they were originally developed, raises issues of validity. How can we assess the appropriateness of reusing the selected submodel in a new context?

The challenge is to define a formal process and framework for agile model-based design reasoning that will enable engineers to elaborate on a question with a high-quality and mixed-fidelity system model that can be configured efficiently from stored submodels. A famework and an information model for modular product modeling based on commercial CAD, CAE, and PDM-technology was proposed in [Sellgren 1999] and further elaborated on in [Sellgren 2003]. This paper presents an approach to question-driven modeling as an enabler of model-based design reasoning. The presented approach is an extension to the information model presented in [Sellgren 1999] with a scenario activity model. Product data management (PDM) technology is used to manage the information objects created and used in the scenario activities. Assessment of the appropriateness of reusing a model is left to the judgment of the user(s), who can find the original context, purpose, and requirements for an existing model or submodel by navigating previous scenarios. In this paper, the scope is restricted to enabling design reasoning on questions that are related to the physical behavior and performance of technical products.

2. Question-driven modeling

Engineering questions are frequently related to a state of knowledge deficit. In general, explicit and reliable answers to such questions cannot be found directly. It is more likely that implicit answers can be found by elaborating on the questions with the aid of a model, that is, through the use of model-based design reasoning. Modeling is a bottleneck and a significant barrier in most model-based engineering approaches. However, it has been shown that a modular model architecture enables efficient and flexible modeling [Sellgren 2003].

The approach to model-based design reasoning presented here is based on a modular model architecture, efficient system model configuration, and capture of the modeling rationale and process as engineering scenarios. A scenario is modeled as a recursive workflow (see the left-hand side of figure 2) with six distinctive steps or scenes:

1. Define the context-dependent engineering problem and reformulate it as one or several question(s). The context may for example be a stored product model of an artifact.
2. Analyze each question and specify the requirements for a target model.

3. Synthesize (i.e., configure) a specific systems model that available knowledge suggests will satisfy the requirements.
4. Perform the simulation, or alternatively a set of simulations in probabilistic design, and store the result as an aggregated information object.
5. Analyze the simulation result to verify and validate (V&V) the model. Store as information object.
6. Synthesize an answer from the analyzed simulation results (and identify new questions).

Each activity in a scenario needs input information, creates output information, is controlled by information, and is performed by an engineer with the aid of a mechanism or tool. The right-hand side of figure 2 shows an IDEF0- or SADT-based [Ross 1977] representation of the six steps, or scenes, in a basic scenario workflow that has been implemented in a research pilot demonstrator. Each of the scenario objects (i.e., the question, model requirements, model simulation result, and answer) has an owner attribute, a state attribute, and a causality relation to the object that was created in the immediately preceding activity in the scenario workflow.

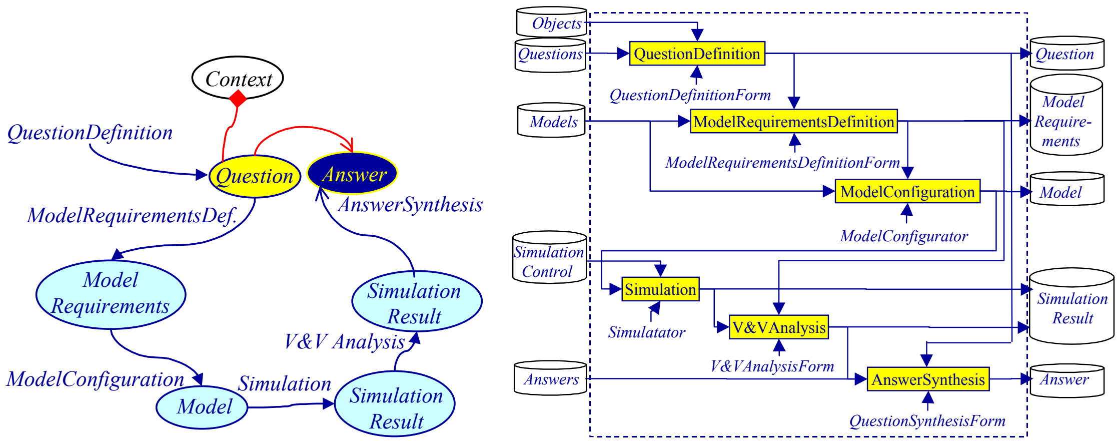


Figure 2. Question-driven modeling (left) and an activity view of a scenario loop (right)

The mechanisms in figure 2 are currently web-based forms, (e.g., one form for question definition and another for model requirements definition), specialized configuration tools, and industry-standard simulator tools (e.g., Ansys for finite element simulation and Adams for multibody dynamic simulations). Questions, answers, model requirements, models, and simulation results are objects that are created, used, and reused in a scenario loop. Figure 3 shows a condensed class diagram of the most important scenario objects.

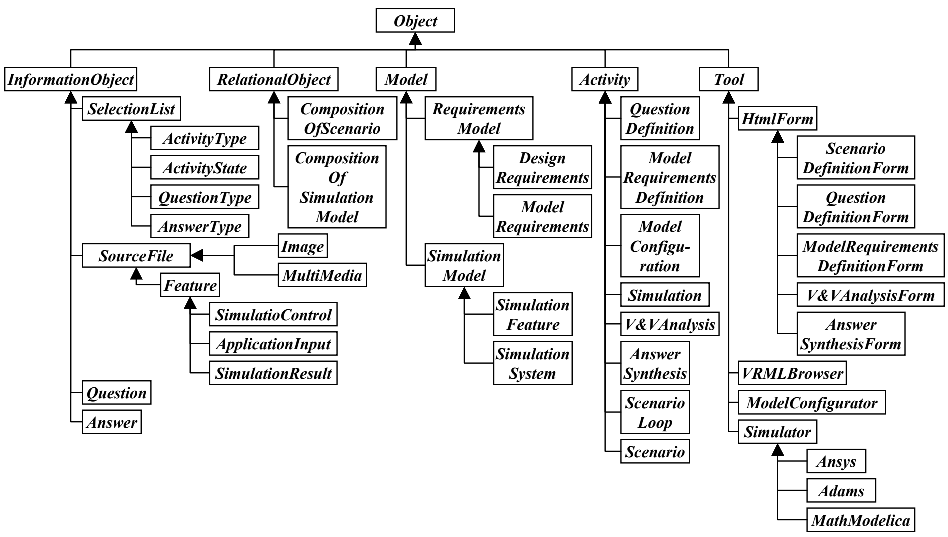


Figure 3. A class-diagram of the most important scenario objects

A *ScenarioLoop* object is an aggregation of the objects created in an elementary scenario loop. A *Scenario* object is a composition of one *ScenarioLoop* object or several serial and/or parallel *ScenarioLoop* objects. All objects are created as Extensible Markup Language (XML) files [W3C 2004] and they have describing attributes and relational attributes that point at other objects that are embodied as XML-files. All files are managed with standard PDM/PLM technology. This approach enables searching of stored questions and answers in a distributed environment and navigation between scenario scenes (e.g., finding the original requirements and the context for a stored model).

A *Question* object has a textural description of the engineering question, a few describing attributes, and, more importantly, a relation attribute which points at a context object (see the example in figure 4). The context object is also a control object for the scenario. A typical context object is a stored design model.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/xsl" href="..\xsl\Question.xsl"?>
<QUESTION>
   <NAME>Question1LiftingUnit</NAME>
   <DESCRIPTION>What is the idealized working envelope
               of the lifting unit concept?</DESCRIPTION>
   <QUESTION_TYPE>Convergent</QUESTION_TYPE>
   <OWNER>Ulf Sellgren</OWNER>
   <QUESTION_STATE>Defined</QUESTION_STATE>
   <CONTEXT TARGET="LiftingUnit(VersionB).xml">Context</CONTEXT>
   <FOCUS TARGET="ModelRequirements1LiftingUnit.xml">Model requirements</FOCUS>
   <ANSWER TARGET="Null.xml">Answer</ANSWER>
   <FEATURE TARGET="Question1LiftingUnitHtml.xml">Feature</FEATURE>
</QUESTION>
```

Figure 4. An XML-file with a defined question object

In general, the model requirements depend strongly on the selected model type and simulator. In the approach presented here, the model type (e.g., finite element model) and target application (e.g., Ansys) are defined in the *ModelRequirementsDefinition* activity and are stored as attributes in a *ModelRequirements* object. A model requirements object has a relation to one question object and a number of descriptive and prescriptive attributes. Values for the prescriptive attributes are selected from predefined tables and include such elements as mechanical, 3-dimensional, quasi-static, linear, deterministic, and hybrid as regards the physical dimensions, spatial dimensions, time dimensions, non-linearity, system randomness, and system fidelity attributes, respectively. The scope of the target systems model is defined by selecting/unselecting parts or submodels [Sellgren 1999] of the context object (i.e., the master *DesignModel*). This is done in the model requirements definition activity. The issue of how to treat the active environment is still open. It is currently treated as two prescriptive attributes, one for boundary conditions and another for working conditions.

ModelConfiguration is the creative act of combining existing submodel variants and using interface features [Sellgren 2003] to connect them in a model that is as simple as possible but still satisfies the model requirements. There is a one-to-many relation between master design model and behavior model variant. In general, non-human systems can only synthesize knowledge from available information in very special cases, which have to be accurately described in advance [Vajna 2002]. Consequently, in the approach presented here, model configuration is treated as a human creative activity, with the engineer assisted by a searching and navigation tool, and a *VRMLBrowser* tool for visualizing models.

3. An industrial case

Volvo Construction Equipment (Volvo CE) is one of the world's leading manufacturers of construction machines, with a product range encompassing wheel loaders, excavators, articulated haulers, motor graders, and more. Construction equipment, such as the wheel loader shown in figure 5, is heavily loaded during normal operation, which gives strength and fatigue predications a high priority. Finite element (FE) analysis currently plays a significant role in life-assessment of critical components at Volvo CE. An identified challenge for the near future is to assist design reasoning by developing models and simulations of various aspects of system performance and multiphysics behavior.

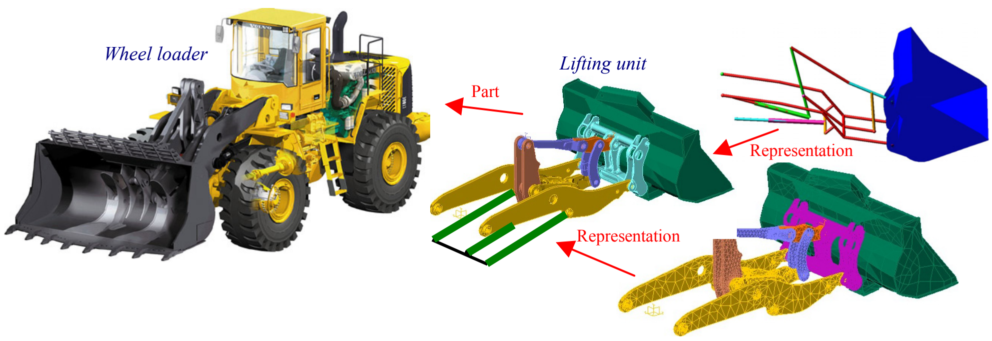


Figure 5. A wheel loader and a lifting unit model with two behavior representations

To address this challenge, existing models of a wheel loader have been decomposed into submodel variants and interface features. Furthermore, the characteristic properties of each submodel and interface feature, and the relations between the objects have been defined as PDM-manageable metadata [Sellgren 2003]. Recently, a scenario metamodel has been defined and evaluated by addressing a number of engineering issues that are related to the physical behavior and performance of a wheel loader lifting unit (see figure 5). The issues have been reformulated as engineering questions and each question has been incorporate into a scenario workflow.

One example of a question that has been addressed is, "What is the dynamic envelope for the bucket CP?" A scenario that addresses this question is shown in figure 6. A stored lifting unit design model was chosen as the context object in the *QuestionDefinitionForm*, see figure 7. In setting the model requirements, the target model was described as *elastic body dynamics*, see figure 7. The prescriptive requirements were defined as *FE* method, *Ansys* target application, *mechanical* physical dimensions, *3-dimensional* spatial dimensions, *quasi-static* time dimension, *deterministic* randomness, *large rotations* non-linearity, *abstract* fidelity, *fixed* boundary conditions, and *composite* working condition. Eight mechanical parts and three hydraulic actuators were chosen from the context design model as the scope of the model configuration.

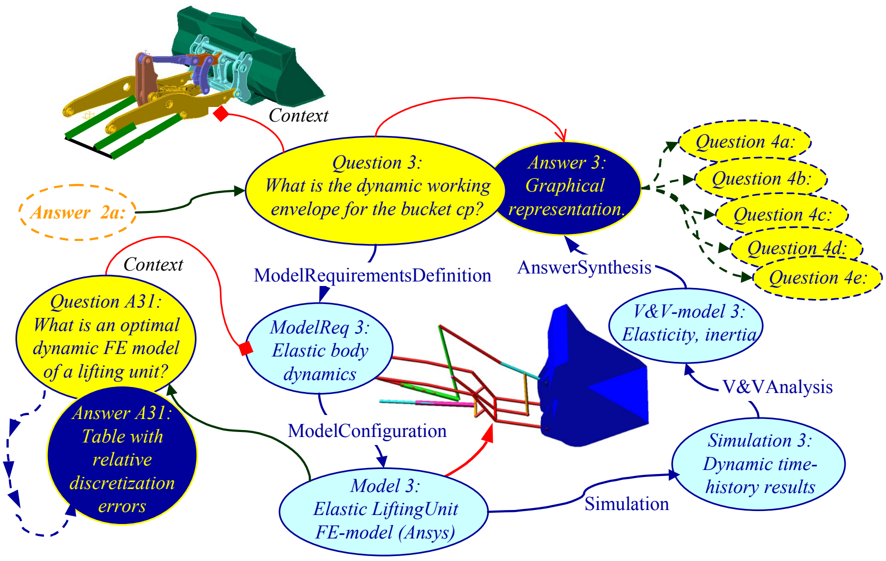
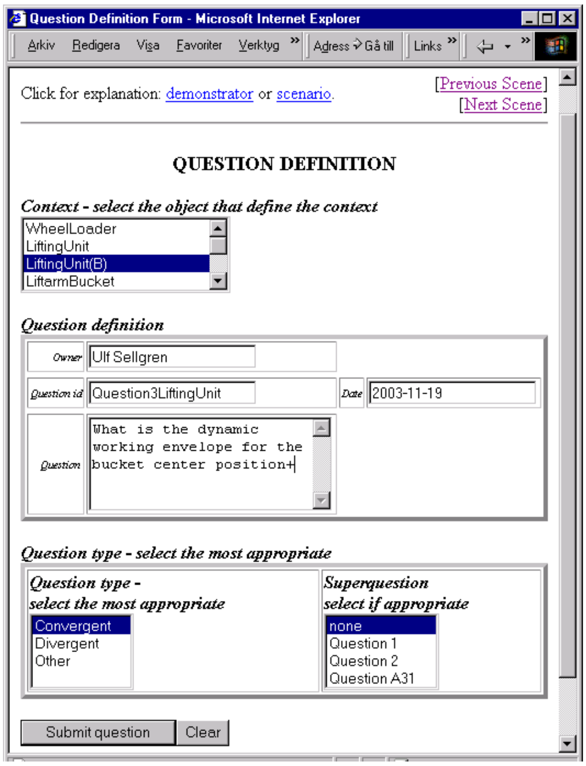


Figure 6. A complex question-driven model-based engineering scenario

Having determined the scope of the submodel, it was possible to identify all the submodel variants that were relevant to the target model as configuration candidates and to make them selectable in a VRML-based model configuration browser (see figure 8). The chosen approach was to connect the submodels that were selected with the configuration browser in a matrix view of the model [Sellgren 2003] (see lower left portion of figure 8).



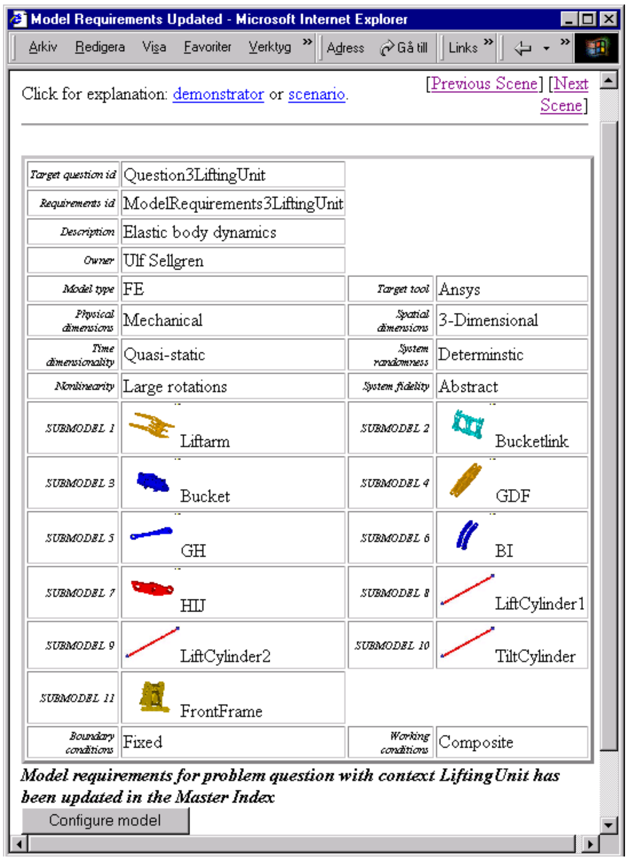


Figure 7. QuestionDefinitionForm (up), and Model Requirements object (down)

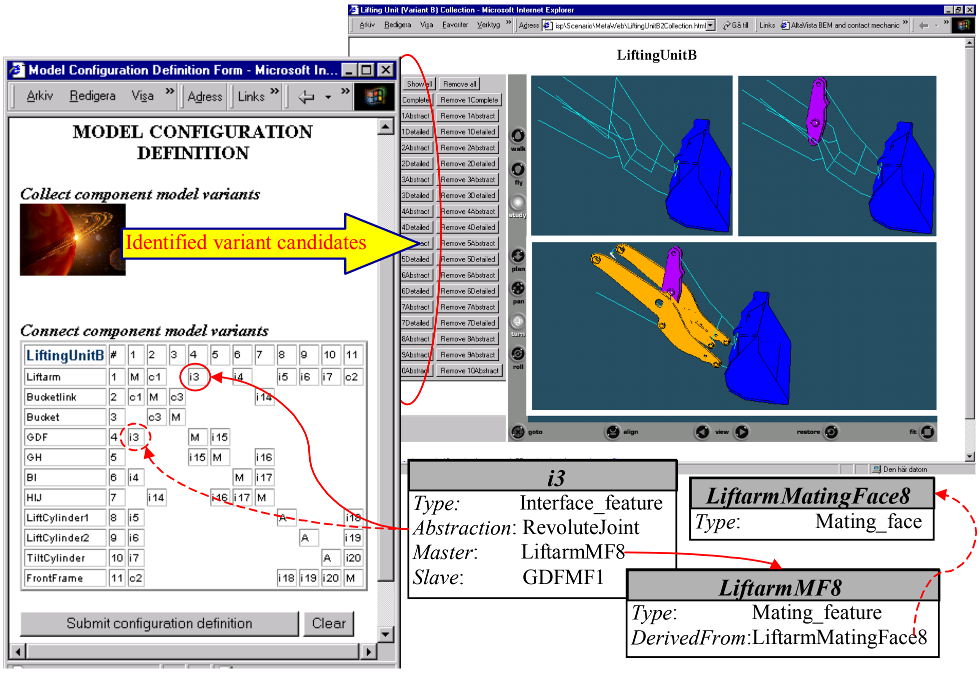


Figure 8. Configuration of a system model from stored submodel variants

The subsequent steps in the scenario are rather straightforward and are not discussed here. Figure 6 indicates that the answer to the question triggers four new engineering questions and, thus, four new model-based scenario candidates. In the model configuration activity, a model performance question is also formulated, with the model requirements as its context object, and an implicit answer is found by navigating stored scenarios. A map of the sequence of engineering issues that have currently been studied is shown as a complex scenario structure in figure 9.

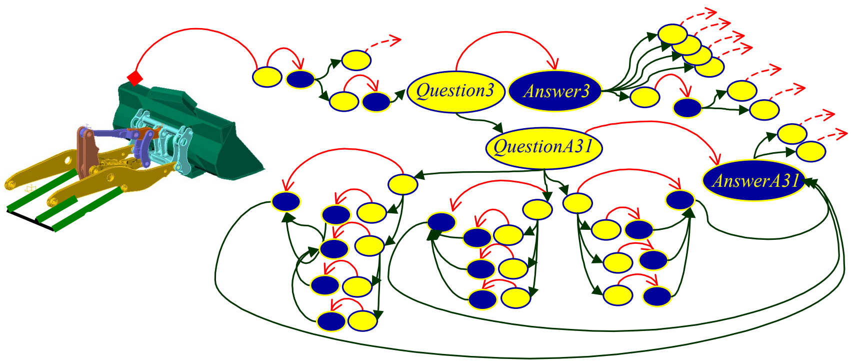


Figure 9. A complex wheel loader scenario with a lifting unit context

4. Conclusions and questions

This paper presents an approach for model-based reasoning about engineering issues related to product behavior and performance. The approach, which is referred to as question-driven modeling, integrates a principle for modular modeling originally proposed in [Sellgren 1999] and a newly developed scenario workflow model.

The workflow model relates modeling and simulation activities and the simulation results to an engineering question and to a context. A stored scenario is a representation of implicit modeling rationale and it can, thus, be viewed as a container of codified (implicit) modeling knowledge.

To assess the pros and cons of the proposed approach, engineering issues from a wheel loader product development case have been reformulated as engineering questions, modeled as a complex scenario for a wheel loader lifting unit, and implemented and tested in a research demonstrator.

Initial tests of the proposed scenario approach to question-driven modeling have been promising and the experience gained from experimenting with this demonstrator of question-driven modeling can be summarized as follows:

- The modular model architecture reduces modeling complexity and significantly reduces the modeling time, especially for modular designs, by making it linear to the total number of submodels and interface objects.
- The modeling and simulation rationale can be found by navigating the relations between question, context, model requirements, and model in the stored scenario object.
- By adding a selectable value for the ScenarioType attribute, it is possible to store also an explicit answer to a question (e.g., a reference to a company standard) as a scenario object.

The conclusions are still tentative because the evaluation results so far have been obtained in an iterative research and implementation loop. A more extensive implementation and evaluation process is under way.

A number of *research questions* that address the usefulness of the presented approach to modeling can also be formulated

- Does navigation of stored scenarios significantly facilitate understanding of design decisions made?
- Does navigation of stored scenarios enable understanding of modeling decisions made?
- Can a novice engineer be trained in modeling by navigating scenarios and created by experienced engineers?
- What is the value of a mechanism that enables answers to formulated design questions to be scrutinized and questioned by others in an organization?
- How can the validity of a formulated answer be assessed?

Acknowledgements

The work presented here was performed in the VISP project and financially supported by the Swedish Agency for Innovation Systems (VINNOVA).

References

Boisot, M.H., "Is your firm a creative destroyer? Competitive learning and knowledge flows in the technological strategies of firms", Research Policy, Vol. 24, pp. 489-506, Elsevier Science B.V, 1995.

Hatchuel, A. and Weil, B., "A new approach of innovative design: an introduction to C-K theory", Proceedings International Conference on Engineering Design – ICED 03, Stockholm, Sweden, 2003.

Ross, D.T., "Structured analysis (SA): A language for communicating ideas", IEEE Transactions on Software Engineering, Vol. SE-3, No. 1, 1977, pp.16-34, 1977.

Sellgren, U., "Simulation-driven design: motives, means, and opportunities", Doctoral thesis, Department of Machine Design, Royal Institute of Technology, Stockholm, Sweden, 1999.

Sellgren, U., "Architecting models of technical systems for non-routine simulations", Proceedings International Conference on Engineering Design – ICED 03, Stockholm, Sweden, 2003.

Troxler P., "Knowledge technologies in engineering design", Proceedings Design 2002, Dubrovnik, Croatia, pp 429-434, 2002.

Tomiyama, T., "Structuring design knowledge for better design synthesis", Proceedings International Conference on Engineering Design – ICED 03, Stockholm, Sweden, 2003.

Ullman, D.G.; D'Ambrosio, B. "Taxonomy for Classifying Engineering Decision Problems and Support Systems", Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, Vol. 9, 5,pp. 427-438, Nov.1995.

Vajna, S, "Approaches of knowledge-based design", Proceedings Design 2002, Dubrovnik,Croatia, pp 93-100, 2002.

W3C,"Extensible Markup Language (XML) 1.0 (Second Edition)", http://www.w3.org/TR/REC-xml, 2004.

Dr. Ulf Sellgren
Royal Institute of Technology (KTH), Department of Machine Design, SE-100 44 Stockholm, Sweden
Telephone: +46-8 790 83 87, Telefax: +46-8 723 17 30
Email: ulfs@md.kth.se