Modelling and Management of Design Artefacts in Design Optimisation*

Arndt Mühlenfeld, Franz Maier, Wolfgang Mayer, and Markus Stumptner

Advanced Computing Research Centre
University of South Australia
Mawson Lakes, SA 5095, Australia
muehlenfeld|franz.maier|wolfgang.mayer|mst@cs.unisa.edu.au

Abstract. Complex design processes often embrace various degrees of virtual development, where complex models and simulations replace traditional construction and testing of physical models. However, as the number of models and their inter-relationships grows, managing processes and models becomes increasingly difficult. We describe how to support product development by applying ontologies to manage and guide the design of simulations and to make domain knowledge readily available though context-specific reuse mechanisms. Based on established engineering standards like ISO 10303 we defined domain-specific abstractions and operators to facilitate information reuse.

Keywords. Design Optimisation, Ontologies, STEP ISO 10303

1 Introduction

While virtualisation has led to considerable streamlining of product design and development processes, the amount of information to be processed through modelling and simulation has grown to an extent where it has become increasingly difficult to manage and analyse models, their underlying assumptions and data produced in different steps of a development process. Hence, tools to support designers and engineers to support their modelling efforts are desired.

In this context, ontologies (i.e., machine interpretable domain concept definitions) allow to capture and query knowledge about models, processes and simulations involved in the development processes. As knowledge becomes explicit (and thereby available for further processing), tools to retrieve and reason about models, simulations and their results permit to exploit information acquired through previous processes to guide the execution of the current development process. In particular in

* This work was funded by the CRC for Advanced Automotive Technology under the IDE project C4-03. We are grateful to Chris Seeling (VPAC) and Daniel Belton (General Motors, Holden Innovation) for providing a test-bed and domain expertise.

multidisciplinary scenarios, where several teams work concurrently to achieve a suitable tradeoff between different requirements, it is important to consolidate information managed in different teams and heterogeneous information systems to provide a consistent picture of the current state of the design processes, artifact(s) and models. Here, we investigate how ontological engineering can improve existing design optimisation scenarios.

We introduce a method of modelling the design process via *task* and *artefact* ontologies to capture knowledge required for subsequent design optimisation. Exploiting captured knowledge, it becomes possible to analyse and compare past modelling efforts with the current analysis scenario to identify potentially redundant simulations and to query and reuse information obtained from previous simulations.

Our work builds on established engineering standards like ISO 10303 [5] and extends existing representations with additional ontologies to provide abstract "views" of concrete development artefacts. Explicit formalisations of essential analysis-specific properties of a design alternative provide the means for isolating and browsing similar "compatible" design optimisation sub-processes. Our framework enables reuse of previous results in design optimisation, hence speeding up the optimisation processes by avoiding unnecessary time-consuming numerical simulations.

In Section 2, the concepts and underlying principles of design optimisation are outlined. Section 3 introduces our approach to ontological process and artefact modelling and discusses the benefits. Section 3.2 presents an example of how we apply ontologies to reuse knowledge required for simulation tasks. Section 4 discusses related work in the area. Section 5 summarises our work and gives an outlook on future research directions.

2 Design Optimisation Process

Multidisciplinary Design Optimisation (MDO) is a form of virtual development where rigorous modelling and optimisation techniques are applied starting early in the design process, to obtain a coarse understanding of different aspects of a design across a number of heterogeneous domains. Rather than optimising each discipline separately, all disciplines are analysed in parallel and the results are merged with the intent to obtain the best design alternative as a compromise of all included disciplines. For example, developing a new car may require a trade-off between engine power, fuel consumption, stability/crashworthiness, and overall weight.

Since modelling and simulation of different aspects may be performed in parallel, on different versions of a design artefact, and at different levels of granularity, ensuring compatibility between models and results is challenging. For example, in the automotive sector, current development practices have resulted in inefficiencies due to inadequate design or simulation models and insufficient data at critical milestones in the stage-gate development process.

Here, formal representations of critical properties of models can help to make explicit assumptions and constraints underlying the individual modelling efforts to detect and help resolve such inconsistencies early. In the following section, we propose an architecture to address this compatibility problem.

3 Building a Design Process Ontology

Proper representation of important properties of design artefacts and simulation processes is vital for reasoning about the design process, prerequisites, and results. While task-specific aspects and concrete execution scenarios can be captured in process ontologies, process representations must be complemented with (abstractions of) artefact models to enable comparison of different analysis tasks and their resulting models. In this paper, we focus on the role of artefact models in our framework; our approach to capture process-related information has been discussed in [7]. The process model is used to specify and capture the execution of simulation tasks and for automated workflow enactment. It also allows to connect engineering decisions and artefacts to the processes that induced them. The inputs and results of the processes are represented as artifact models. To relate different executed processes, it is necessary to compare the corresponding process elements and artifact representations in both processes. In the following, it is assumed that the artefact models have been annotated with provenance information to capture process-related information.

To effectively address these issues, machine-processable representations of design artefacts must be created that support comparing different artifacts with respect to specific criteria. Detailed tool-specific representations of artefacts are directly available from modelling tools (for example, *CATIA*), but those models are not easily utilised in a wider context due to proprietary representations. This issue has largely been overcome by increased adoption of standards like STEP [5] that have established standardised ontologies and representations for artifacts in particular application domains. Therefore, we base our modelling efforts on the STEP Application Protocols (APs) as reference representation for concrete artefact models.

While concrete artefact models, such as detailed geometry information in CAD documents, capture the artefact in full detail, these representations are often too fine-grained to draw useful inferences. Instead, a more abstract representation is desired that captures only those properties that are relevant for a given analysis task. For example, to assess whether two results obtained from finite-element simulations of different parts of a vehicle body may be assembled into a larger model, the detailed geometry information may not be relevant, but the material properties and version number of the simulation software are critical. Similarly, essential requirements and assumptions underlying individual analysis sub-processes can be aggregated into abstract representations (we call those *"views"*).

Since different analysis tasks are likely to address different aspects of a product, a single view is unlikely to be sufficient. Furthermore, a monolithic representation may also hinder reuse, since irrelevant attributes may affect the compatibility test. Hence, abstractions and compatibility criteria between abstract models must be defined with respect to particular analysis goals.

Here, STEP/EXPRESS has been identified as suitable framework, since it provides an expressive language that can capture not only concrete representations of artifacts through APs, but also formalise the aggregation operators to transform concrete representations into abstract views and to define compatibility between abstract views. The powerful representation in EXPRESS has shown to lead to simpler development through abstract specification and meta programming [2].

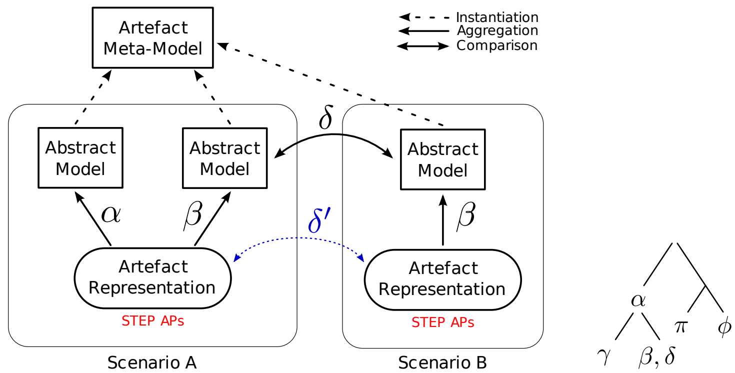

Fig. 1. Artefact model architecture and operator ontology (conceptual)

We use meta-modelling to consolidate different models and compatibility criteria. The meta-models integrate artefact and process models with the domain models of the STEP Standard [5]. We use EXPRESS as the formalism for design artefact representation as well as for specification of concrete and abstract artefact properties and their mappings. Furthermore, the STEP standard comprises a large body of standardised, rigorously defined APs. We can directly access the vast amount of domain knowledge already formalised in STEP, by also expressing the engineering model in EXPRESS. Furthermore, the models on the meta-level form a frame of reference for all concrete models and thereby provide the means to express the relations between properties on different levels of abstraction.

3.1 Aggregation of Design Artefact Properties and Meta-Data

We distinguish models at different layers of abstraction. The STEP APs constitute the basis for all subsequent model transformations. Concrete documents representing instances of APs produced by design and engineering tools are depicted as ovals in Fig. 1. For example, the STEP integrated resources (represented by parts 41–58) provide constructs for geometric and topological representation as well as mathematical descriptions and numerical analysis. Elements of the simulation process can be described by ISO 10303. We assume a domain model that utilises parts of AP 214 and covers the analysis results including the geometry given in a STEP physical file format (Part 21), parametrisation and constraints defined in STEP Part 108, and the finite element mesh defined in Parts 107 and 104. Execution-specific information such as time stamps, machine identifiers, number of CPUs involved, and different software versions can be directly obtained by extension of core APs.

By aggregation from information in the concrete documents, more abstract views are created that capture different aspects of a design that have been identified relevant for a given analysis task. The resulting abstract models no longer reflect all engineering details, but are easier to store and manipulate. Each view conforms to an abstract meta-model that specifies the language that is used to describe a view. In our framework, STEP/EXPRESS is used to define both the meta-model and views.

Similarly, a specification of the current analysis scenario (Scenario B in Fig. 1) is also represented as abstract view. Hence, the goal of comparing past simulations and their results to the current scenario reduces to the problem of obtaining and comparing suitable abstract views. By comparing abstract views (using function δ in Fig. 1) that reflect only relevant properties, compatibility of concrete artifacts (relation δ' in Fig. 1) can be assessed.

Abstractions from concrete models are also defined in EXPRESS (functions α and β in Fig. 1). By applying transformation operations specified at the meta-level between the meta-models, essential properties of a concrete model are computed to obtain its abstract counterpart. For example, transformation α in Fig. 1 could represent an abstraction from the concrete shape of a geometry model (as specified in STEP Part 42) to yield an abstract model that represents only the extent and mass of the described object. This transformation would be specified as operation on documents conforming to STEP Part 42 to yield an instance of the meta-model describing the extent+mass model. Let the latter be denoted as *ExtentMassModel*.

To relate different abstract views to each other, comparison operators are defined for each abstract view. Each operator is a function that ascertains whether two instances of given views satisfy particular compatibility criteria. In Fig. 1, operation δ represents an abstract comparison function that assesses whether two instances of *ExtentMassModel* are compatible with respect to a specific analysis. Note that δ' is difficult to compute directly, since that entails specifying precise comparison operators on varying structures directly in the language defined by the STEP APs. By applying aggregation beforehand, the computation can be simplified considerably. Furthermore, the separation of abstraction and comparison aids the dynamic adaption of transformation and comparison operators based on either operand – a technique that would be difficult to achieve when using the detailed representations.

Since abstractions and compatibility criteria depend on the analysis under consideration, an ontology of transformations is defined (depicted in the right part of Fig. 1) that is used to select suitable criteria given a desired analysis. Hence, through selection of suitable aggregation and comparison operations, different abstract views of the same concrete artifact or simulation result can be obtained and compared with the current analysis setting. For example, an ontology of operators may specify that for fuel consumption calculations, two artifacts with equal overall mass are considered equivalent without considering the precise geometric shape. Hence, a suitable abstract comparison operation δ can be chosen, together with abstraction functions to automatically compute the abstract models from the concrete artifact models.

The successive aggregation of models leads to a formal representation that can be easily stored in a design repository for subsequent query evaluation and reuse. By adoption of automated reasoning technology and consolidation of design artefact properties, results of simulation scenarios can be compared and existing MDO data sets can be selected for reuse.

3.2 Example

In the following, we illustrate how to utilise the approach for reuse of design knowledge in context of an analysis in the automotive design domain.

Setup. Assume that an MDO task is carried out in order to optimise the deformation of a vehicle's front part in response to a crash impact. Design objectives are high energy absorption but low mass. The shape optimisation problem is investigated while the remaining car body remains unchanged. Different approaches to optimisation are possible: e.g., a morphing approach that keeps the original finite element model and only modifies the coordinates of the nodes, or, an approach based on re-meshing of the existing finite element model. As the intended modifications of the vehicle's front part are minor, an approach with an unchanged finite element model is sufficient. A corresponding parametric model represented in STEP/EXPRESS is given from which a concrete domain model is generated for given model parameters.

Domain Model. The representation of the domain model is achieved with STEP AP214 *Core data for automotive mechanical design processes* covering data model and requirements of the chosen design discipline. AP214 provides a detailed model of design artefacts including for example the geometry and the parametric model. As a top data model of AP214 the *Application Integrated Model* (AIM) makes use of STEP *Integrated Generic Resources* (parts 41-58) for its development. The AIM of a STEP AP is finally used for data exchange.

STEP parts, such as Part 52, the *mesh-based topology*, and Part 49, the *process structures and properties*, are used to describe the finite element mesh of the vehicle's front part, and for representation of particular process properties. The geometry of the investigated design artefact can be represented with Part 42.

After an optimisation task has completed, the created files are collected and represented in EXPRESS. Geometry, mesh and further results created during the optimisation process are stored in a repository, together with meta-information about the performed simulation. Here, maximising energy absorption while keeping the mass of the design part low and meta-information about the experiment such as for example *Timestamp*, *MachineID*, *CPUInformation*, *OperatingSystem*, *SoftwareVersion*, *Duration*, etc. are used as additional information besides the data model. This information can also be represented in the EXPRESS modelling language and therefore can be integrated with the meta-model for the design artefacts.

Top-Down Reuse. Considering the design of an MDO task, a designer typically selects from different categories of criteria to define the setup of the simulations to be performed. In addition to design objectives, involved disciplines, design variables and constraints, the process steps and involved software systems to execute the scenario have to be defined.

MDO task ontologies can serve as meta-models of the optimisation process steps [7]. To make a decision which design artefacts can be selected from the design repository for reuse, firstly, comparison operators have to be developed to connect the constraints and objectives formulated on a high level of abstraction with the properties attached to design artefacts on the domain layer. To formally define these comparison operators between design artefacts, dependencies between input variables and output responses of previous runs have to be analysed.

For example, to maximise the energy absorption and minimise the mass of a vehicle's front part, input variables that are most significant for these design objectives are derived from previous runs. Hence, the abstraction and comparison operators from Section 3.1 need not be static, but can vary between design scenarios.

Summary. The example sketches different categories of design optimisation problems and the necessity of a formalism to compare design artefacts for the purpose of simulation reuse. It illustrates the role the STEP Application Protocols play in this context. Aggregation and abstraction of design artefact properties defined in EXPRESS enables to derive abstract design attributes from a concrete domain. The process is aligned with a meta-model for design artefacts. When retrieving information about existing MDO simulation runs, the requirements provided by the designer are decomposed and traced down to a concrete design repository. Our approach allows the definition of a unified model that supports automated execution as well as a high level view used in organisational decision making in a consistent framework.

4 Related Work

The use of ontologies in design, engineering and process modelling is widespread; hence, the following discussion focuses on selected representative works. Ontologies have been applied in engineering contexts, resulting in several proposals to capture distributed design knowledge [3] and to provide interoperability at the semantic level (as opposed to data structure based data exchange) [6, 8].

The Performance Simulation Initiative (PSI) [9] used ontologies to express dependencies between concurrent activities in dynamic engineering design processes in order to speed-up design cycles through increased concurrency between activities. Different to our work, the focus is on improving process structures; hence, design artifacts are modelled using very abstract ontologies that do not permit the detailed comparison and reuse that we aim to address.

Work on model-based interoperability [1] aims at merging EXPRESS and ontologies based on formal logic. While this approach is desirable to apply the strong inference capabilities of Description Logics to EXPRESS models, further research is necessary to extend the current mapping to include complex constraints.

Our approach consolidates process and artefact ontologies under a common STEP/EXPRESS meta-model. We chose EXPRESS due to existing artefact ontologies described with the standard, and its suitability for meta-modelling. Other standards, for example the *Process Specification Language* (PSL), may seem better suited for process modelling. However, PSL lacks support for context relationships, unexpected activity outcomes and needs better definitions of process artefacts [4].

5 Conclusion and Future Work

A number of ontologies have been developed to support designers in annotating and browsing design variants, but most approaches require custom-built interfaces and are not well-integrated in existing engineering environments. Our approach to extend the established STEP standard with domain-specific abstractions aims to avoid this limitation by building upon well-established domain ontologies to define and implement a framework to extract relevant properties from concrete domain models produced by actual engineering tools. We have illustrated that transformations within a common meta-model framework make possible to intelligently reuse partial results stored in a common model repository, at a level of detail that has exceeds the capabilities of previous work. We acknowledge that our approach may require significant modelling effort to represent all desired processes, but believe that the use of STEP and established standards helps to considerably lower this barrier. Furthermore, the approach can be applied incrementally, limiting the impact on the overall design process.

Based on a case study drawn from the automotive industry, we have begun to isolate and formalise relevant properties and relationships of engineering models, with the immediate goal of transforming, executing and monitoring optimisation processes on top of publicly available execution platforms, replacing the current proprietary implementation. Further work includes to extend the set of properties that currently considered in our models, and to evaluate our approach using the case study and other application scenarios.

References

1. C. Agostinho et al. EXPRESS to OWL morphism: making possible to enrich ISO10303 modules. In *Proceedings of ISPE CE 2007*, pages 391–402, 2007.
2. Y. Ait-Ameur, F. Besnard, P. Girard, G. Pierra, and J. C. Potier. Formal specification and metaprogramming in the EXPRESS language. In *Proc. SEKE'95*, pages 181–188, 1995.
3. R. Fruchter and P. Demian. CoMem: Designing an interaction experience for reuse of rich contextual knowledge from a corporate memory. *AI EDAM*, 16(3):127–147, 2002.
4. A. G. Gunendran, R. I. Young, A. F. Cutting-Decelle, and J. P. Bourey. Organising manufacturing information for engineering interoperability. In *Proc. IESA 2007*, pages 587–598, 2007.
5. ISO. *10303-11:1994: Part 11: The EXPRESS language reference manual*. ISO, 1994.
6. A. Maier, H.-P. Schnurr, and Y. Sure. Ontology-based information integration in the automotive industry. In *ISWC*, volume 2870 of *LNCS*, pages 897–912. Springer, 2003.
7. F. Maier, W. Mayer, M. Stumptner, and A. Mühlenfeld. Ontology-based process modelling for design optimisation support. In *Proc. DCC'08*, 2008.
8. P. H. P. Nguyen and D. Corbett. Building corporate knowledge through ontology integration. *Advances in Knowledge Acquisition and Management*, 4303/2006:223–229, 2006.
9. R. Sohnius et al. Managing concurrent engineering design processes and associated knowledge. In *ISPE CE*, pages 198–205. IOS Press, 2006.