# Using Ontologies to Optimise Design-Driven Development Processes[*]

Wolfgang Mayer, Arndt Mühlenfeld, and Markus Stumptner

Advanced Computing Research Centre, University of South Australia
Mawson Lakes Blvd, Mawson Lakes, SA 5095, Adelaide, Australia
mayer|arndt.muehlenfeld|mst@cs.unisa.edu.au
Fax: +61 8 8302 3988

**Abstract.** While optimisation-driven design has become prevalent in many engineering disciplines, support for designers to effectively *use* the results of simulation processes has not been addressed satisfactorily, since, processes monitoring and reuse of simulation results are not well-integrated into current development practices. We introduce a framework to integrate Multidisciplinary Design Optimisation processes using ontological engineering, where artefact and simulation models are exploited to yield more effective optimisation-driven development. We show how meta-modelling techniques can overcome representational and semantic differences between analysis disciplines and execution environments.

**Keywords.** Ontological Engineering, Process Modelling, Design Optimisation

## 1 Introduction

In the design and engineering context, ontologies provide an explicit formalisation of design knowledge that is otherwise distributed among several teams [6]. Ontologies also aid in semantic interoperability between design disciplines due to the introduction of meta-models that serve as a linking element between disciplines [12], providing means to reason about process-, simulation- and domain-specific aspects [3].

As designs become more complex, designers and engineers increasingly rely on tool support to manage not only design artefacts, but also the design processes themselves. In order to store, manipulate, connect and validate processes, semantic representations of processes are desired that are able to convey not only the structure, but also the semantics of process parameters and activities unambiguously.

While research in distributed scientific computing [11] and Web Services has led to numerous attempts to represent the "meaning" of individual tasks, most frameworks do not fully address the challenges posed by non-trivial design tasks, where

[*] This work was funded by the CRC for Advanced Automotive Technology under project 10 Integrated Design Environment. We are grateful to Chris Seeling (VPAC) and Daniel Belton (General Motors/Holden Innovation) for providing a test-bed and domain expertise.

information is rich in structure and detailed interpretation of semantics is necessary. Conversely, static artefact models fail to capture dynamic aspects of processes and their execution. Hence, effective tool support requires an *integrated* approach that facilitates the development of adequate models of *individual artefacts and tasks*, *processes* and their *execution* [13].

This paper extends the emphasis from artefact-centric ontologies and process-oriented approaches to a comprehensive unified framework. We outline an ontological representation of typical multidisciplinary optimisation processes within analysis-driven design processes. In this context, ontologies may serve as part of a reusable framework in which reasoning about artefacts, related processes and simulation tasks and their results is exploited to design, enact and adapt product development processes. Although not explored in this paper, the integration of different models and disciplines may also prove useful to guide designers and engineers in navigating analysis results and in exploring design alternatives. We are not primarily concerned with establishing interoperability between engineering applications, but focus on how existing analysis tool chains can be employed more effectively.

We illustrate how ontologies of processes and engineering models can be combined in a uniform framework build on top of well-established engineering standards (STEP/EXPRESS [8]) and how declarative mappings between meta-models facilitate automated synthesis and adaption of complex processes. Our work aims to unify dynamic process enactment, detailed representations of design-related artefacts, and expressive knowledge-representation formalisms to gain a more powerful framework that surpasses the limitations of purely process-oriented and concept-oriented representations. While no single language is likely to satisfy all use cases, the integration of models at the *meta-level* can be done using a small set of powerful languages. In this paper, we advocate the use of EXPRESS to formalise the necessary meta-models and model transformations, since this language has been shown to be powerful enough to express and execute model mappings within a unified framework [15].

In Section 2, the concepts and requirements underlying model-driven engineering processes are outlined. In Section 3 our approach to ontological process modelling is introduced and the role of ontologies in design scenarios is discussed. Section 4 presents the architecture of our framework and Section 5 discusses our approach to overcome disparities between different models. Section 6 elaborates on possible benefits of our framework. Section 7 summarises our view on knowledge modelling in the design and engineering context and outlines future research directions.

## 2 Concurrent Design Optimisation Processes

The desire to reduce costs and product cycles and increasing product complexity have led to a paradigm shift towards virtualisation. Early adoption of standardised modelling and simulation techniques and frequent model interchange strive to eliminate the need for physical artefacts.

Multidisciplinary Design Optimisation (MDO) is a form of virtual development where rigorous modelling and optimisation techniques are applied early in the design process, to obtain a coarse understanding of different aspects of a design across a number of heterogeneous domains. Disciplines are analysed in parallel and the results are merged to obtain the design alternative representing the "best" compromise.

A prerequisite for this is the representation of MDO *processes and information flow* in a way that permits semantic analysis. For this we have to focus on two views: the traditional product/design modelling view, and the explicit modelling of the process view. We use what we call an Ontology-based approach, where "ontology" is used in the interpretation of [9] as meaning a set of concepts plus logical axioms that describe their interrelations.

## 3 Ontologies for Engineering Processes

Adequate representations of structure, function and semantic annotation are essential requirements for reasoning about design artefacts as well as design processes, their prerequisites and their results.

Ontologies provide the means to represent the relationships between artefacts and sub-artefacts, as well as (material and domain-specific) properties and annotations made by designers/engineers, in a way that is amenable to semantic analysis and translation. Standards like STEP [8] and ontologies to represent function [10] serve as a starting point for the specific purpose of artefact modelling.

The process aspects are concerned with the representation and execution of design and engineering processes within and across organisational boundaries. Here, a consistent formal process model allows to connect engineering decisions and artefacts to the processes that induced them.

Different formalisations of semantic representations of activities in design and engineering processes have been proposed, but do not satisfy all requirements of the MDO context: Models proposed for process flows [1], workflow and Web service execution typically provide means to specify preconditions and effects, but leave the precise ontology to be used unspecified or suffer from limited expressiveness. Similarly, process modelling efforts in the scientific [11] and engineering domains [16] do support dynamic and adaptive processes, but fail to embrace detailed representations of artefacts and information manipulated by these processes; typically, design artefacts are represented as black boxes. In the MDO context, both process enactment and detailed artefact knowledge are required for successful process management. Conversely, well-established standards like STEP are designed to capture detailed artefact information, but are not specifically designed for managing dynamic processes. We aim to address to narrow this gap by introducing an ontological framework that combines dynamic process adaption mechanisms with detailed engineering models to complement existing approaches in order to support the design and execution of MDO analyses.

## 4 Ontology Architecture

The approach to design support sketched in this paper relies on formal ontologies to represent processes and design knowledge, as well as automated reasoning tech-

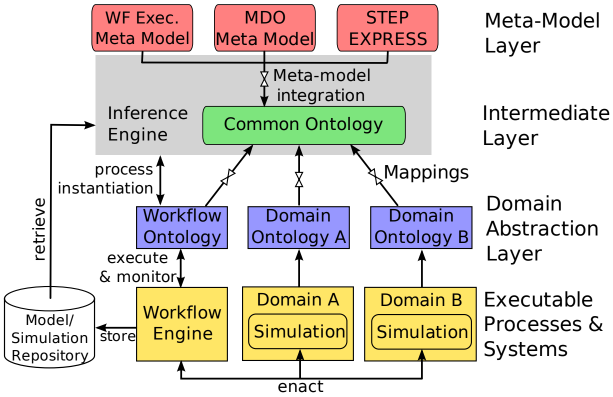

Fig. 1: Ontological Support Framework

niques to build, validate, and translate design processes and design knowledge. However, an evaluation of well-known languages and ontologies showed that no single ontology or language is sufficient to represent all desired aspects [13]. Hence, different representations must be combined into a unified framework to overcome representational differences.

The explicit use of *adaptors* [2] has been advocated to bridge gaps between ontologies, including for parametric configuration problems [5]. Although not based on adaptors, [10] show that domain-specific ontologies representing the function of devices can be related via a common model to extend reasoning about roles and functions beyond a single domain. Our approach can be seen as an attempt to adopt the multiple-viewpoints paradigm proposed in [7], but in a *declarative* framework.

We rely on domain experts and knowledge engineers to identify and represent relevant interactions between domain ontologies and formalise mappings between ontologies (Fig. 1). A layered approach where meta-models are located at the top, unified task and artefact ontologies comprise the intermediate layer, and domain-specific ontologies form the bottom layer in the ontology hierarchy has been adopted. Concrete executable systems, such as CAD environments, optimisation tools and workflow orchestration engines, are located below the knowledge representation layers. The model and simulation repository acts as a conceptual memory component where ontologies, processes and information about (partially) executed simulations (obtained from the execution platform) and their results are stored. This repository is accessed to draw inferences based on historic process information.

From analysis of individual domains, ontologies of domain-specific concepts, properties and relations are created. Process execution environments, for example workflow enactment systems, are treated in the same way. As a result, a set of domain ontologies is obtained. Domain-independent aspects and processes are found by generalisation of domain-specific ontologies to form the intermediate layer: By defining suitable ontology mappings, domain-specific knowledge is mapped into the unified common ontology at the intermediate level. Hence, it becomes possible to describe and reason about domain-independent and task-independent concepts, such as

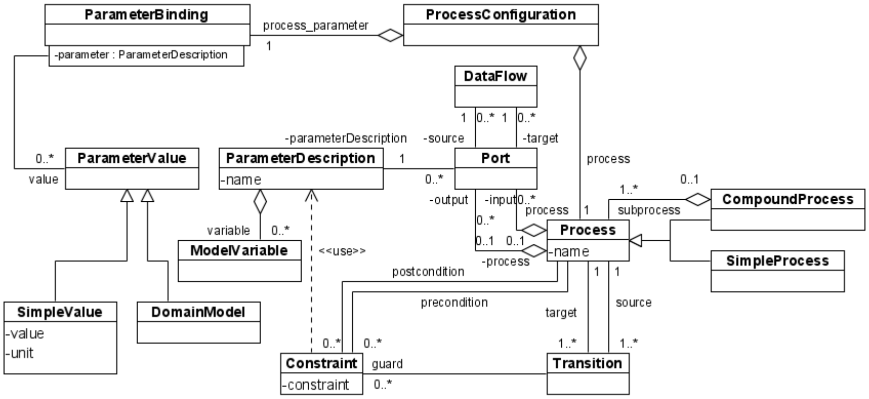

Fig. 2: MDO Process Meta-Model [13]

execution traces and execution histories. Common ontologies allow to *design*, *trace*, *reason about* and *execute* analysis-driven processes. Support environments developed for the design and execution of distributed scientific experiments have demonstrated that this is feasible in certain domains while hiding much of the underlying formal knowledge representation mechanisms from engineers [11].

Ontologies and inference systems that comprise the intermediate layer serve as a platform to integrate information and processes obtained from different domains and expressed in languages defined by different ontologies. This aspect is vital when dealing with artefact and process information simultaneously, which may not be expressed in a common language. Differences between ontologies, workflow languages and data formats can be overcome at the meta-model level [4].

## 5 Integration by Declarative Abstraction

Translation between the intermediate layer and the ontologies below is accomplished by adaptors that map between domain-independent and domain-specific representations. The same idea is used to instantiate abstract process models at the intermediate level to generate process specifications tailored to specific workflow execution environments, such as [11], to leverage mature platforms for process enactment.

We created a generic meta-model [13] of optimisation-driven design processes (Fig. 2) to form the basis for the joint semantic representation . The central components are *Process*es, *DataFlow* and *ParameterDescription*s, which represent (concrete or abstract) tasks, information flow between tasks, and the structure and meaning of passed data from an engineer's point of view, respectively. The model is generic in that bidirectional translations to and from widely-adopted process modelling languages can be defined. The abstract representation of parametric engineering models as *Parameter*s and *DomainModel*s allows to introduce detailed domain-specific engineering models through specialisation; *Constraint*s serve to specify and record additional semantic relationships expressed by extensions of well-established STEP Application Protocols (APs).

The representation of domain models and tasks via abstract parameters and constraints is the foundation of our engineering support framework: Critical properties of concrete artefact models and simulation tasks are captured as domain- and analysis-specific abstractions of concrete artefacts and simulations formalised as ontologies represented in a uniform language based on STEP/EXPRESS. The common representation allows to define relationships between different abstract models, such as equivalence criteria with respect to a given analysis task, which are used to find and retrieve stored optimisation scenarios compatible with the current problem under consideration. The retrieved information may subsequently be used to guide further design optimisation and exploration processes. The relevance of STEP/EXPRESS in this context is reinforced by the fact that domain-specific models of design artefacts conforming to standardised APs can be obtained directly from engineering tools, such as CAD systems. Hence, abstract representations of design artefacts can be derive automatically once given abstractions have been specified. Furthermore, APs serve as the domain-specific reference ontologies upon which abstractions are built using declarative knowledge representation paradigms, reducing the effort to build abstractions and increasing the chance of reusing mappings in other projects. Our meta-model is an abstraction of concrete process models used in a particular domain, which in turn are specifications of possible execution scenarios that may occur. The model must be instantiated for a particular domain to obtain domain-specific process models, which are subsequently used to analyse, execute and monitor processes.

Analogous to the synthesis of execution models through ontology mappings, complex computing environments can be described by ontologies that allow to automatically create a workflow to store and access required information, considering the individual capabilities of existing sub-systems. Hence, it becomes possible to supplement certain system components with additional databases holding (meta-)data that cannot be processed by legacy components.

## 6 Supporting Design and Engineering

The architecture introduced in the previous sections facilitates the support of a variety of different modelling and engineering tasks. For space reasons we restrict our presentation to a few representative examples.

The main benefits relevant for designers and engineers are as follows: **(a)** Through model management enabled by our framework, each analysis may be based on a consistent and up-to-date global view of the design process. **(b)** Tracing the evolution of designs and analysis results becomes possible. In particular, the origin and assumptions underlying critical models and parameters can be recorded and later exploited to guide analysis efforts. For example, if it is known that a design constraint on a particular artefact part is only a default value, it may be opted to re-negotiate the exact value rather than spending much effort on satisfying the constraint (which may become obsolete in later design stages). **(c)** System support for selection, composition and reuse of analysis processes may help to avoid errors in the setup stages and to shorten turnaround time. Flexible integration of different systems through automatic workflow synthesis and orchestration lessens the burden of integrating disparate systems from the IT personnel's point of view.

**Process execution.** Using automated reasoning technology, process models can automatically be translated into workflow enactment models that are subsequently executed. This allows to automatically monitor, store, and reason about process outcomes that are handled by the MDO environment. Simulation inputs and results can be compared and possible changes to the process can be suggested and validated. Declarative ontology mappings enable to separate modelling and reasoning from implementation-dependent aspects, legacy applications and scientific Grid platforms.

**Simulation reuse.** MDO optimisation tasks can be adapted and streamlined if suitable results are available from previous similar analysis. Using ontologies to compare past simulations to the current analysis, reuse of partial results to streamline the current simulations becomes possible. Conceptually, our approach extends the idea of process refinement described in the context of distributed scientific workflows [14] to complex engineering models that cannot be described using the simple meta-data approach provided by well-known Web services and Grid platforms.

**Optimisation design.** Similar reasoning can be applied in the preparation stage *before* optimisation processes are executed. For example, formal process descriptions can be used to ensure that a suite of experiments leads to compatible results that can subsequently be aggregated into a global view. In addition, potentially inappropriate process inputs can be flagged by comparing parameter settings chosen by engineers with formal models of successful and failed executions stored in the repository.

**Error handling.** Simulations may also benefit from improved robustness of models and execution through semi-automated error recovery. If a simulation aborts due to modelling errors or out-of-range input values, formal ontologies and a repository of models and execution traces can help to determine whether a different model is available that does not exhibit the same problem. Since permissible input values may not be known explicitly for complex models, case-based or machine learning approaches can be adopted to build the necessary descriptions incrementally.

## 7 Discussion and outlook

We presented a meta-model-driven framework for the execution and analysis of optimisation-driven engineering processes, where the integration of common and domain-specific ontologies allows to reason about process executions and simulation results on a meta-level. We concede that the initial effort to create ontology mappings may be considerable for large projects, but believe that the potential benefits of the approach by far outweigh its costs. By relying on existing engineering standards, such as the STEP APs and other data representations, the modelling effort can be directed towards ontology mappings at the meta-model layer, rather than on the lower-level ontologies and data format conversion implementation. To further curb the complexity underlying the creation of models, a hierarchy of meta-models is proposed, where domain- and tool-specific models at the lower layers are reconciled into higher-level generic models that allow to translate between semantically overlapping domains. The same approach is applied to mediate between heterogeneous infrastructure components.

Currently, we explore these technologies for consistency assessment of processes and their instances based on a crash simulation scenario in the automotive industry, and investigate translation of higher-level representations into different workflow representations. Extension and detailed evaluation on additional scenarios and design domains remains future work. Creating mappings between ontologies to complex (common and domain-specific) entities and relations and to explore different inference mechanisms to support engineers in defining, maintaining and applying such transformations are additional topics we are planning to investigate.

## References

1. C. Bock and M. Grueninger. PSL: A semantic domain for flow models. *Software Systems Modeling*, pages 209–231, 2005.
2. B. Chandrasekaran, J. Josephson, and R. Benjamins. The ontology of tasks and methods. In *Proc. KAW'98*, 1998.
3. C. Dartigues and P. Ghodous. Product data exchange using ontologies. In John S. Gero, editor, *Proc. (AID'02)*, pages 617–637. Cambridge, 2002.
4. Laurent Deshayes, Sebti Foufou, and Michael Gruninger. An ontology architecture for standards integration and conformance in manufacturing. In *Proc. IDMME*, 2006.
5. Dieter Fensel, Enrico Motta, Stefan Decker, and Zdenek Zdráhal. Using ontologies for defining tasks, problem-solving methods and their mappings. In Enric Plaza and V. Richard Benjamins, editors, *EKAW*, volume 1319 of *LNCS*, pages 113–128, 1997.
6. A. Gómez-Pérez, M. Fernández-López, and O. Corcho. *Ontological Engineering*. 2004.
7. John Haymaker, John C. Kunz, Ben Suter, and Martin A. Fischer. Perspectors: composable, reusable reasoning modules to construct an engineering view from other engineering views. *Advanced Engineering Informatics*, 18(1):49–67, 2004.
8. ISO. *10303-11:1994: Part 11: The EXPRESS language reference manual*. ISO, 1994.
9. M. Kifer, G. Lausen, and J. Wu. Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42(4):741–843, 1995.
10. Yoshinobu Kitamura, Yusuke Koji, and Riichiro Mizoguchi. An ontological model of device function: industrial deployment and lessons learned. *Applied Ontology*, 1(3–4):237–262, 2006.
11. Bertram Ludäscher et al. Scientific workflow management and the kepler system. *Concurrency and Computation: Practice and Experience*, 18(10):1039–1065, 2006.
12. Andreas Maier, Hans-Peter Schnurr, and York Sure. Ontology-based information integration in the automotive industry. In Dieter Fensel, Katia P. Sycara, and John Mylopoulos, editors, *ISWC*, volume 2870 of *LNCS*, pages 897–912. Springer, 2003.
13. Franz Maier, Wolfgang Mayer, Markus Stumptner, and Arndt Mühlenfeld. Ontology-based process modelling for design optimisation support. In *Proc. DCC'08*, 2008.
14. Simon Miles et al. Connecting scientific data to scientific experiments with provenance. In *Proc. E-SCIENCE'07*, pages 179–186. IEEE Computer Society Press, 2007.
15. Vincent Plantec, Alain AND Ribaud. PLATYPUS : A STEP-based integration framework. In *14th Interdisciplinary Information Management Talks (IDIMT-2006)*, 2006.
16. Richard Sohnius et al. Managing concurrent engineering design processes and associated knowledge. In Parisa Ghodous, Rose Dieng-Kuntz, and Geilson Loureiro, editors, *ISPE CE*, pages 198–205. IOS Press, 2006.