Paper:
Crick, T., Dunning, P., Kim, H. & Padget, J. (2009). Engineering design optimization using services and workflows. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 367*(1898), 2741-2751.
http://dx.doi.org/10.1098/rsta.2009.0035

Engineering Design Optimisation using Services and Workflows

By Tom Crick[a], Peter Dunning[b], Hyunsun Kim[b], Julian Padget[a]

[a]*Department of Computer Science, University of Bath*
[b]*Department of Mechanical Engineering, University of Bath*

Multi-disciplinary optimisation (MDO) is the process whereby the often conflicting requirements of the different disciplines to the engineering design process attempt to converge upon a description that represents an acceptable compromise in the design space. We present a simple demonstrator of a flexible workflow framework for engineering design optimisation using an e-Science tool. This paper provides a concise introduction to MDO, complemented by a summary of the related tools and techniques developed under the umbrella of the UK e-Science program that we have explored in support of the engineering process. The main contributions of this paper are: (i) a description of the optimisation workflow that has been developed in the Taverna workbench, (ii) a demonstrator of a structural optimisation process with a range of tool options using common benchmark problems, (iii) some reflections on the experience of software engineering meeting mechanical engineering (iv) an indicative discussion on the feasibility of a "plug-and-play" engineering environment for analysis and design.

Keywords: multi-disciplinary optimisation, web services, workflows, semantic web

1. Introduction

(*a*) *The Engineering Perspective*

Complex engineering design is realised through the combined efforts of a number of specialist design teams with discipline-specific skills, tools and knowledge. The competitive environment of the engineering industry demands an optimum design through continuous improvement in performance and economy of design. However, due to the interdependent nature of disciplines, achieving an optimal design can be difficult and slow as each specialist team often lacks a direct understanding of the global consequences.

Multi-disciplinary design optimisation (MDO) has been defined as "methodology for the design of systems in which strong interaction between disciplines motivates designers to simultaneously manipulate variables in several disciplines" (Sobieszczanski-Sobieski and Haftka, 1997). Research in MDO has received increasing interest as it aims to reconcile potential conflicts by treating the design as a whole, taking account of the interdependency of design disciplines. Two categories of methods have emerged: single-level and multi-level formulations. The former typically uses a single optimiser which is applied to the entire multi-disciplinary system. Whilst this approach may be considered the more intuitive, the typical size and highly complex nature of the problems are usually unsuitable for a single optimiser. Conversely, the multi-level approach decomposes a design problem. Both analysis and optimisation are carried out for each sub-system and the interaction between sub-systems are considered by a system-level optimiser to determine the solution (Martins and Marriage, 2007; Yi et al., 2008).

In the modern engineering environment, the multi-level formulation is often the preferred approach. There are several reasons for this, but two key factors are (i) the decomposition of a large problem into a number of sub-system optimisations is naturally suited to parallel and distributed computing, and this can significantly reduce computational costs to realistic levels, and (ii) as the structure of engineering organisations typically reflect disciplines and each discipline team works largely independently, the multi-level approach is somewhat akin to the current design industry, and hence more suited to industrial practice.

The multi-level formulation considers the highly-coupled inter-dependency of multi-disciplinary design criteria at system level. Therefore, it follows that the optimum solution can be sensitive to the system architecture and implementation (Alexandrov and Kodiyalam, 2007; Brown and Olds, 2006). There have been limited studies that provide a generalised understanding of various methods and their performance, and the MDO implementations have been specific to design problems and applications. As the design evolves and the tools are modified, it requires a significant modification to the MDO framework and the suitability of the modified architecture for the given problem cannot be guaranteed.

Additional difficulties arise in the changing environment of engineering industry, where designs are increasingly carried out in various geographical locations on heterogeneous platforms. This calls for a more flexible and easy-to-understand MDO system for integrating both legacy codes and proprietary software with a range of model representations and fidelity (Giesing and Barthelemy, 1998).

(*b*) *From e-Science to e-Engineering*

A key aim of e-Science activity has been to ease accessibility to data and computational resources, wherever they might be, and however they might be described. In technological terms this translates to workflow design and enactment on the one hand, and service description and discovery on the other. Active research initiatives on service discovery can be seen particularly in the chemistry and bio-informatics communities: for example, BioMoby at `www.biomoby.org` (retrieved 20081212) offers interoperability between biological data hosts and analytical services, Biocatalogue at `www.biocatalogue.org` (retrieved 20081212) provides a curated catalogue of Life Science Web Services. The myGrid project at `www.mygrid.org.uk` (retrieved 20081212) has demonstrated how bio-informatics research can be assisted (Oinn et al., 2004) through the automated identification of Web services and a visual programming environment—the Taverna toolkit (Oinn et al., 2006)—for the construction and execution of workflows. Association of semantic information with services and its subsequent discovery is being enabled by tools such as the semantics-enabled extension of UDDI, Grimoires (Fang et al., 2008), and the general purpose, extensible brokerage framework, Knoogle (Chapman et al., 2007)—both OMII projects, see `www.omii.ac.uk` (retrieved 20081212)—while Seekda at `www.seekda.com` (retrieved 20081212) offers a text-based searcher for web services and the Feta Lord et al. (2005) component of Taverna is developing a semantic search interface. In earlier work, we made some steps towards the semantic description of mathematical services (Caprotti et al., 2004), which is particularly relevant for engineering, but in many fields effective domain-specific description is an emerging topic.

The objective of the work reported here is to investigate the applicability of the Taverna workbench to a classical engineering optimisation problem. We chose to use Taverna

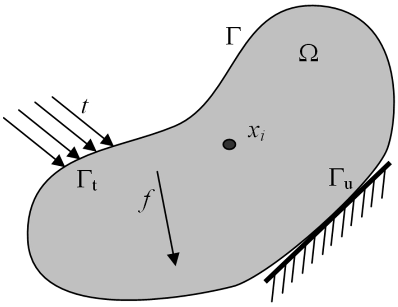


Figure 1: Generalised two-dimensional continuum

because it has been used for a diverse range of domains inclusing medicine, astronomy, social science and music, as well as bioinformatics. It is clear from the degree of up-take that the use of tools to support the authoring and enactment of workflows has been beneficial to many different communities, by (i) enabling access at a distance to resources, and (ii) allowing the researcher to focus on how to combine those resources. There are many parallels to be found in the engineering design process: (i) workflows capture common, even standardised, processes (ii) resources are not necessarily co-located, and (iii) much tedious effort is spent copying and transforming data output by one analysis as input for another analysis. It is notable that commercial software packages are starting to exploit the workflow concept in their interfaces but, by being essentially closed systems, have the effect of locking users into particular products.

By building a system based on workflow, we are able to delegate file manipulation and house-keeping details to the enactment engine. Furthermore, from an engineering design perspective, it is now feasible to deploy many more analysis codes and with equal ease integrate them into workflows and visualise the results. We now proceed to discuss the proof-of-concept demonstrator.

2. Services and Workflows

(*a*) *Structural optimisation algorithm*

The design problem considered for the demonstrator study is a material distribution problem in a continuum such as Figure 1. The continuum domain is represented by Ω enclosed by a boundary, Γ , with body forces, f, the boundary traction, t on $\Gamma_t \in \Gamma$ and support on $\Gamma_u \in \Gamma$. The design variable, x represents the existence of material which is either present or absent, $x \in \{0, 1\}$, as shown in Figure 1.

The formulation of the optimisation problem is to minimise the total compliance subject to the equilibrium and volume constraints. The implementation typically employs the existence of finite elements as design variables. A common approach to this discrete problem is to relax the design variables to $0 < x \leq 1$, but to penalise the intermediate values by power-law (Bensøe, 1995). The optimisation problem can therefore be written as:

$$\text{Min } C(x) = \sum_i x_i^p \mathbf{u}_i^T \mathbf{k} \mathbf{u}_i \text{ subject to } \mathbf{KU} = \mathbf{F} \text{ and } V(x) \leq V_0$$

where C denotes total compliance; $\mathbf{u}_i$ and $\mathbf{k}$ are the elemental displacement and stiffness, respectively; p is the penalisation power; $\mathbf{K}$ is the global stiffness matrix; $\mathbf{U}$ is the global

```
input  : a problem independent component $\mathsf{PI}$
input  : a problem dependent component $\mathsf{PD}$
input  : optimisation process control parameters $\mathsf{PO}$
output : an optimised problem dependent component
var    : the value of $\mathsf{PD}$ at step $i$, denoted $\mathsf{PD}_i$
var    : the value of $\mathsf{PD}$ at step $i-1$, denoted $\mathsf{PD}_{i-1}$
var    : the value of $\mathsf{PD}$ at step $i-2$, denoted $\mathsf{PD}_{i-2}$
var    : the strain energy of the model, denoted $\mathsf{SE}$
$\mathsf{PD}_i \leftarrow$ Setup ($\mathsf{PD}$,$\mathsf{PO}$)
$\mathsf{PD}_{i-1} \leftarrow \mathsf{PD}_i$
$\mathsf{PD}_{i-2} \leftarrow \mathsf{PD}_i$
repeat
    $\mathsf{SE} \leftarrow$ Analyse ($\mathsf{PI}$,$\mathsf{PD}_i$,$\mathsf{PO}$)
    $\frac{d\mathsf{SE}}{dx}, \frac{d^2\mathsf{SE}}{dx^2} \leftarrow$ Sensitivity-analysis ($\mathsf{PO}$,$\mathsf{SE}$,$\mathsf{PD}_i$)
    $\mathsf{PD}_{i-2} \leftarrow \mathsf{PD}_{i-1}$
    $\mathsf{PD}_{i-1} \leftarrow \mathsf{PD}_i$
    $\mathsf{PD}_i \leftarrow$ Optimise ($\mathsf{PD}_i$,$\mathsf{PD}_{i-1}$,$\mathsf{PD}_{i-2}$,$\mathsf{PO}$,$\mathsf{PI}$,$\frac{d\mathsf{SE}}{dx}$,$\frac{d^2\mathsf{SE}}{dx^2}$)
until Converged? ($\mathsf{PD}_i$,$\mathsf{PD}_{i-1}$,$\mathsf{PO}$)
return $\mathsf{PD}_i$
```

Algorithm 1: The structural optimisation process

displacement vector; **F** is the force vector; V is the total volume of the design; V_0 is the specified volume limit for the solution.

The optimisation problem can be characterised by several parameters: (i) structural, that are the material properties, boundary conditions, loading and design domain size, and (ii) optimisation, that are the penalisation power, volume fraction and convergence criteria.

Optimisation begins by discretising the design domain continuum using finite elements and the design variables x are the continuous variation of the existence of an element, sometimes referred to as artificial density of an element. A finite element analysis is usually employed to compute the nodal displacements, which in turn are used to determine the sensitivities required for optimisation. The optimisation uses the Method of Moving Asymptotes (MMA) optimiser (Svanberg, 1987), which works by updating the elemental artificial densities. This process is repeated until the convergence criterion is met. This process is formalised in Algorithm 1.

(*b*) *Workflow construction*

Taverna can discover and utilize both local programs and deployed web services as components in workflows. All the components reported here were published as web services and are thus potentially re-usable by others. A Taverna workflow that implements Algorithm 1 is shown in Figure 2. The algorithm and the workflow evolved iteratively through a process of collaborative authoring, each functioning as a boundary object between the domains of the participants, and resulting in the identification of five major components for the workflow: (i) design initialisation (ii) analysis of the current design (iii) sensitivity analysis (iv) optimisation and design update, and (v) the convergence test. For the analysis step, three finite element analysis programs were available:

1. CFE, an “in-house” legacy C code which was designed specifically to undertake the analysis step of the optimisation process. This is deployed as a web service using the gSOAP toolkit (van Engelen and Gallivan, 2002) and demonstrates the capacity to publish C/C++/FORTRAN codes as web services.
2. A commercial package, ANSYS, as an example of proprietary software with a license requirement (ANSYS 11.0SP1). To incorporate ANSYS into the workflow a macro was written in the ANSYS Parametric Development Language (APDL) that executes the required analysis. Additionally, a pre-processing step—implemented as a Taverna shim (see §4.4 of Oinn et al. (2006)), that is a service whose purpose to carry out some minor operation to establish compatibility between two other services—converts the input data file into the format required by ANSYS. The interface is generated by the Soaplab2 (Senger et al., 2003, 2008) web service deployment tool and demonstrates the creation of services from command-line driven engineering analysis tools.
3. A MATLAB script (MATLAB r2003a). The script was specifically written to execute the required analysis. This service is also deployed using Soaplab2 and demonstrates the means to publish services built on widely-used engineering scripting software.

All the analysis programs accepted the same input files with the same format and produced output files in a consistent format, making them completely interchangeable. Each analysis program was limited to a two dimensional static linear elastic analysis of a rectangular domain of square elements of varying density. The inputs and outputs for all workflow components are summarised in Table 1. All data are in plain text file format except for the convergence test output, which is a binary number.

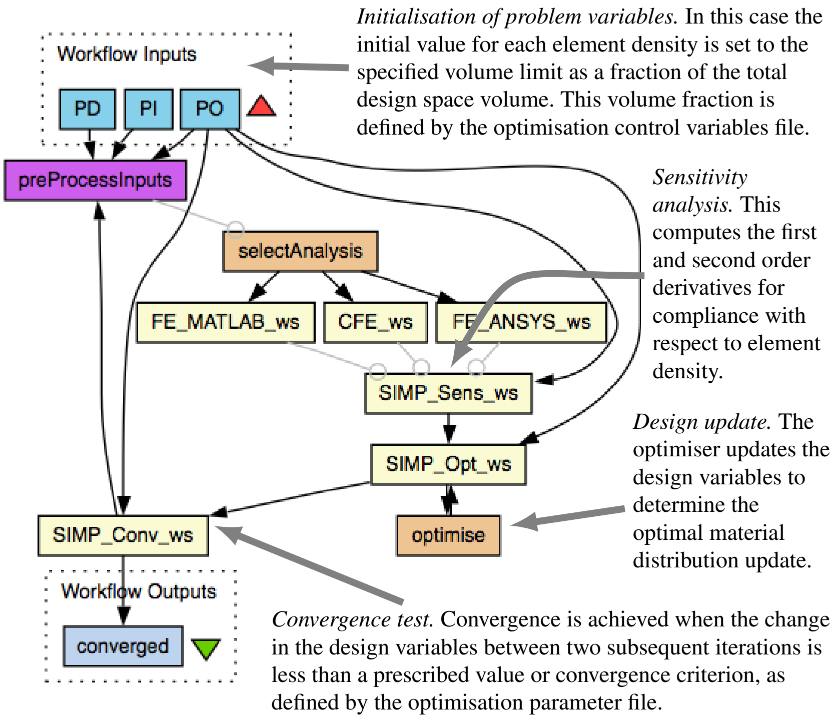
3. Results

In this section we use two popular structural benchmark problems and solve them using the optimisation workflow described in the previous section. The two problems are a short cantilever beam and a MBB beam (Bensøe, 1995). Each structure was discretised with square elements of unit area. Both optimisation problems were run using all available analysis programs and the optimisation parameters were the same for both problems: $p = 3$; $V_0 = 0.4$; Convergence criteria = 0.01.

Figure 3 (left) depicts the structural design environment for the popular cantilever beam of aspect ratio 1.6, with one edge clamped and a central vertical load applied on the other side. The optimum solution was obtained after 42 iterations as shown in Figure 3 (right). This is typical of the solutions obtained in existing literature, thus validating the optimisation algorithm implemented as a workflow. The second test case is the MBB beam, which is a simply supported beam of aspect ratio 6 with a central vertical load. The results obtained were in each case the well-known optimum solution.

4. Discussion and Related Work

We have built and validated—using some well-known benchmark structures—a proof-of-concept workflow demonstrator that addresses or facilitates the points raised in Giesing and Barthelemy (1998)—specifically flexibility, provenance, multiple (consistent) models, distribution and resource brokerage—by re-deploying tools conceived for e-Science, to enable more flexible and intuitive MDO processes that allow for: (i) the continuing use of favoured legacy code and prototype scripts as well as commercial software in a common



Initialisation of problem variables. In this case the initial value for each element density is set to the specified volume limit as a fraction of the total design space volume. This volume fraction is defined by the optimisation control variables file.

Sensitivity analysis. This computes the first and second order derivatives for compliance with respect to element density.

Design update. The optimiser updates the design variables to determine the optimal material distribution update.

Convergence test. Convergence is achieved when the change in the design variables between two subsequent iterations is less than a prescribed value or convergence criterion, as defined by the optimisation parameter file.

Figure 2: The structural optimisation workflow in Taverna

framework and (ii) the necessary variation in model representation and precision. It is interesting to observe that ten years on from the above paper, Moore et al. (2008) make similar observations and call for the development of an open-source framework for MDO.

A notable difference between our work and several examples in the current literature on workflow in MDO is our use of an off-the-shelf framework, allowing for concentration on developing programs and wrappers, instead of developing the framework itself, as is the focus of Wang et al. (2003); Hao et al. (2004); Kim et al. (2006); Lähr and Bletzinger (2007). Working with an existing community of (bio-informatics) users, clearly underpins claims for usability and even longevity, as well as the capacity for more rapid growth, given an attractive set of services and workflows, in new domains.

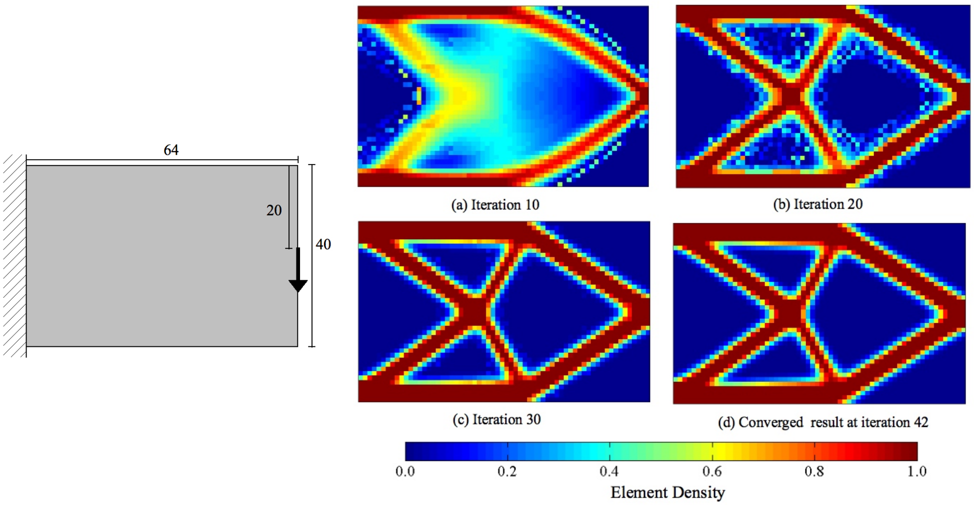
Our planned next steps include investigation of the use of standard workflow descriptions—Taverna uses its own workflow language, called SCUFL, rather the industry-standard BPEL (Oasis, 2007), so workflows are currently not portable—the development of semantic descriptions of the components, and undertaking a wider range of case studies.

(a) *Technical Issues*

Commercial software packages, such as Engineous, Noesis and Phoenix offer similar facilities for workflow creation, visualisation and management, although in each case the user is effectively limited to using the components provided with the package. The real benefit of using a workflow engine like Taverna is the relative ease of utilising web services, potentially leading to: (i) management and use of proprietary analysis tools by con-

Component	Input	Output
Design variable initialisation	Structural parameters Optimisation parameters	Initial element densities
FE Analysis	Structural parameters Optimisation parameters Element densities	Element compliance
Sensitivity analysis	Element densities Element compliance Optimisation parameters	1st order derivative 2nd order derivative
Optimiser (MMA)	Structural parameters Element densities Densities from previous 2 iterations Element compliance 1st and 2nd order derivatives Optimisation parameters MMA variables from previous iteration	Current MMA variables Updated element densities
Convergence test	Updated element densities Previous element densities Optimisation parameters	Convergence result (1 = converged, 0 = continue)

Table 1: Workflow component inputs and output



(a) Iteration 10

(b) Iteration 20

(c) Iteration 30

(d) Converged result at iteration 42

Figure 3: Initial design domain of short cantilever beam (left) and its optimization (right)

tractors, integrating them into the MDO workflow, minimising risk of loss of intellectual property, reducing design time and improving quality control over (sub-contracted) components (ii) sharing "in-house" codes with external users instead having to act as a software provider and maintainer, as well as allowing straightforward version control (iii) capacity for the automatic capture of component provenance information and potential for full life-cycle knowledge management.

We foresee a particular benefit arising from the use of web services for the engineering research community. There has been significant effort into the development of alternative MDO methods, but few attempts at comprehensive comparison *between* methods. One reason is the amount of work required to develop and implement the sizeable range of available methods on a common platform, as most MDO methods for practical problems tend to be designed only for specific applications (Martins and Marriage, 2007; Alexandrov and Kodiyalam, 2007). A possible solution is the concurrent publication of the article *and* its implementation as a web service—an approach along these lines has been implemented by the London Mathematical Society's Journal of Computation and Mathematics for several years, for example see `http://www.lms.ac.uk/jcm/11/lms2007-056/`. Some recent publications in the MDO literature have reached similar conclusions to ourselves about the desirability of workflow approaches, see for example Shi et al. (2005); Bereneds et al. (2008); Moore et al. (2008), but we observe that the published descriptions appear to use bespoke software rather than workflow tools.

A notable drawback of accessing commercial analysis tools as services, is that it bypasses much of their value-add functionality, such as post-processing and visualisation. This factor is also identified by Oinn et al. (2006) in their extensive analysis of Taverna for life sciences applications and holds true for the engineering sector as well.

(*b*) *Reflections on Process*

Bringing together engineering codes and workflow software has been a learning process for both parties. Apart from the initial challenge of appreciating and understanding each other's vocabulary, there have been deeper-rooted issues around the advantages (or otherwise) of bringing in another layer of software technology and the adaptation of legacy code for the new environment.

There have been several situations during this work that might be thought of as "cultural" issues, but not being sociologists, these observations should be seen as purely anecdotal and without significant foundation. One aspect that surprised the computer scientists was the apparently low importance given to making software re-usable: modifications were proposed to make components work in the particular context of use, with relatively little consideration of new environments. Put another way: aspects of engineering practice are now common-place in computer science (software engineering), but these principles have not necessarily made their way back to software development in engineering.

Much of the literature on cultural issues in engineering has addressed ethnicity or organisational factors rather than domain discipline. However, Bond and Ricci (1992) explored the ways in which different disciplines work together in the context of aircraft design. Interestingly, the conclusions they reached resonate equally well with our experience of computer scientists working with mechanical engineers. We paraphrase and summarise their conclusions here and comment upon them: (i) an inter-disciplinary project proceeds through the cooperation of specialists; (ii) each specialist has its own model (or models) of the design for various purposes—we used a shared model (the algorithm) to communicate with one another; (iii) specialists have limited ability to understand each other's models—we now each have a limited understanding of each other's domain; (iv) design proceeds by successive refinement of the models, which are coordinated and updated together—indeed: we prototyped the workflow and the algorithm and revised and updated them together; (v) the design decisions, which are acts of commitment and model refinement, are negotiated by the specialists among themselves.

Finally, over and above the practical issues identified in §4.a, there is a qualitative aspect that is enabled by the adoption of web services and workflow. Much of design optimisation has been based on the parametric representation defined at the initial design stage. This restricts the solution space and prevents optimisation methods from exploring *all* potential solutions. As the design matures and the scale increases to higher levels of detail, more refined analysis and optimisation methods are required and results from the previous stages and legacy systems do not always translate well, both requiring many hours of manual process and potential loss of information. Early design decisions in one discipline may be challenged as the problem is better understood, but the consequences of change for other disciplines are less understood, thus it is simpler to remain at the local optimum. Furthermore, the selection of codes and numerical tools leading to local attractors, may as much be a function of economic and social factors as technical suitability. However, the accessibility of a wide range of codes, capture of provenance information and the ease of trying out alternative design avenues, would ease the exploration of multiple design spaces, and offers the chance to make a notable step forward in multi-disciplinary optimisation.

Acknowledgements. The authors would like to thank: (i) Prof K Svanberg for the implementation of the MMA algorithm, (ii) the Numerical Analysis Group at the Rutherford Appleton Laboratory for the Fortran 77 HSL package MA57, and (iii) Ian Dunlop and Stian Soiland-Reyes of the myGrid team for their help with Taverna via the taverna-users mailing list. (iv) This work was partially funded by the EPSRC Bridging the Gaps program, through the project "Creativity in Systems Design: the Bath Interactive Ideas" (EP/E018122/1).

References

N.M. Alexandrov and S. Kodiyalam. Initial Results of an MDO Evaluation Survey. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 2007. AIAA-1998-4884.

M.P. Bensøe. *Optimization of Structural Topology, Shape and Material.* Springer Verlag, 1995.

J.P.T.J Bereneds, M.J.L van Tooren, and E.J Schut. Design and Implementation of a New Generation Multi-Agent Task Environment Framework. *Proceedings of the 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2008.

A.H. Bond and R.J. Ricci. Cooperation in Aircraft Design. *Research in Engineering Design*, 4(2): 115–130, June 1992. DOI: 10.1007/BF01580149.

N.F. Brown and J.R. Olds. Evaluation of Multi-disciplinary Optimization Techniques Applied to a Reusable Launch Vehicle. *Journal of Spacecraft and Rickets*, 43:1289–1300, 2006.

O. Caprotti, M. Dewar, J.H. Davenport, and J. Padget. Mathematics on the (Semantic) Net. In Christoph Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *Proceedings of the European Symposium on the Semantic Web*, volume 3053 of *LNCS*, pages 213–224. Springer Verlag, 2004. ISBN 3-540-21999-4.

Neil Chapman, Simone Ludwig, William Naylor, Julian Padget, and Omer Rana. Matchmaking Support for Dynamic Workflow Composition. In *Proceedings of 3rd IEEE International Conference on eScience and Grid Computing*, pages 371–378. IEEE, December 2007. DOI:10.1109/E-SCIENCE.2007.48.

Weijian Fang, Simon Miles, and Luc Moreau. Performance analysis of a semantics-enabled service registry. *Concurrency and Computation: Practice and Experience*, 20(3):207–223, 2008.

J.P. Giesing and J.M. Barthelemy. A Summary of Industry MDO Applications and Needs. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998. AIAA-1998-4737.

Qi Hao, Weiming Shen, Seong-Whan Park, Jai-Kyung Lee, Zhan Zhang, and Byung-Chun Shin. An agent-based e-engineering services framework for engineering design and optimization. In

IEA/AIE'2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence, pages 1016–1022. Springer Springer Verlag Inc, 2004. ISBN 3-540-22007-0. doi: http://dx.doi.org/10.1007/b97304.

Hongman Kim, Scott Ragon, James Mullins, and Jaroslaw Sobieszczanski-Sobieski. A web-based collaborative environment for bi-level integrated system synthesis (bliss). In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 1–10, 1-4 May 2006.

André Lähr and Kai-Uwe Bletzinger. Agent-based distributed and interdisciplinary collaboration and analysis in heterogeneous engineering environments. In *7th World Congress on Structural and Multidisciplinary Optimization*, pages 637–646, 21-25 May 2007.

Phillip W. Lord, Pinar Alper, Chris Wroe, and Carole A. Goble. Feta: A light-weight architecture for user oriented semantic service discovery. In Asunción Gómez-Pérez and Jérôme Euzenat, editors, *ESWC*, volume 3532 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 2005. ISBN 3-540-26124-9.

J.R.R.A. Martins and C. Marriage. An Object-Oriented Framework for Multi-Disciplinary Design Optimization. In *Proceedings of the 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2007. AIAA-2007-1906.

Kenneth T. Moore, Bret A. Naylor, and Justin S. Gray. The development of an open source framework for multidisciplinary analysis and optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 1–13, 10-12 September 2008.

Oasis. Web services business process execution language version 2.0. `http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html`, 2007. Retrieved 20081205.

T. Oinn, M. Greenwood, M. Addis, M. N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. R. Pocock, M. Senger, R. Stevens, A. Wipat, and C. Wroe. Taverna: Lessons in Creating a Workflow Environment for the Life Sciences. *Concurrency and Computation: Practice and Experience*, 18:1067–1100, 2006.

Thomas Oinn, Matthew Addis, Justin Ferris, Darren Marvin, Martin Senger, Mark Greenwood, Tim Carver, Kevin Glover, Matthew Pocock, Anil Wipat, and Peter Li. Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–3054, November 2004. ISSN 1367-4803. doi: 10.1093/bioinformatics/bth361.

Martin Senger, Peter Rice, and Tom Oinn. Soaplab - A Unified Sesame Door to Analysis Tools. In *Proceedings of the UK e-Science All Hands Meeting (AHM2003)*, pages 509–513, 2003. URL `http://www.nesc.ac.uk/events/ahm2003/AHMCD/pdf/115.pdf`.

Martin Senger, Peter Rice, Alan Bleasby, Tom Oinn, and Mahmat Uludag. Soaplab2: More Reliable Sesame Door to Bioinformatics Programs. In *Proceedings of the 9th Bioinformatics Open Source Conference (BOSC 2008)*, 2008. URL `http://www.open-bio.org/w/images/3/3c/Rice_Soaplab2_abstract.pdf`.

D. Shi, J. Yin, G. Chen, and J. Dong. A Workflow-Based Distributed Multi-Disciplinary Design Optimization Framework. In *Proceedings of the 9th International Conference on Computer Supported Cooperative Work in Design*, volume 1, pages 497–502, 2005.

J. Sobieszczanski-Sobieski and R.T. Haftka. Multi-Disciplinary Aerospace Design Optimization: Survey of Recent Developments. *Structural Optimization*, 14:1–23, 1997.

K. Svanberg. The Method of Moving Asymptotes – A New Method for Structural Optimization. *International Journal for Numerical Methods in Engineering*, 24, 1987.

Robert van Engelen and Kyle Gallivan. The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks. In *2nd IEEE International Symposium on Cluster Computing and the Grid (CCGrid2002)*, pages 128–135. IEEE Computer Society, 2002. URL `http://www.cs.fsu.edu/~engelen/ccgrid.pdf`.

Y. D Wang, W. M Shen, and H. Ghenniwa. Webblow: a web/agent-based multidisciplinary design optimization environment. *Computers In Industry*, 52:17–28, 2003.

S.I. Yi, J.K. Shin, and G.J Park. Comparison of MDO Methods with Mathematical Examples. *Structural and Multidisciplinary Optimization*, 35:391–402, 2008.

List of Figures