5th International Conference on Experiments/Process/System Modeling/Simulation/Optimization
5th IC-EpsMsO
Athens, 3-6 July, 2013


# DESIGN OF AN ONTOLOGY FOR SIMULATION WORKFLOW OPTIMIZATION

**Moussas V.C.[1,2], Tsahalis J. [1], and Tsahalis H.-T.[1]**

[1] Paragon S.A.
Protopapadaki 19, Galatsi, GR-11147, Athens, Greece
e-mail: jtsahalis@paragon.gr, web page: http://www.paragon.gr

[2] School of Technological Applications, Tech. Educ. Inst. (TEI) of Athens,
Ag. Spyridonos Str., Egaleo 12210, Athens, Greece
e-mail: vmouss@teiath.gr, web page: http://users.teiath.gr/vmouss/

**Keywords:** Optimization; ontology; owl; simulation workflow; product development process;

**Abstract.** *This paper investigates the development of an optimization ontology model. The optimization ontology assists in exchanging semantic project information when working with optimization problems and ontology end users are developers sharing domain knowledge as well as instance knowledge of the optimization methods and tools. This work investigates the development of a General Optimization Ontology (GOO) ontology that will be designed and structured with main focus on the optimization domain but will be also capable to support any other application domains with a minor expansion. The authors developed the basic concepts common to all optimization problems that are required by the core part of the ontology. The first aim is to support automatic selection of the appropriate optimization tool for a given optimization problem, then it will be defined how that could be complemented by domain specific modules tailored to the optimization problems under investigation. The use case examples used to test the ontology focus on the simulation workflow optimization, with emphasis to the use of heterogeneous applications in heterogeneous environments.*

## 1 INTRODUCTION

Ontologies provide a way to establish common vocabularies and capture domain knowledge and they can be used to improve search, integration of heterogeneous information, browsing or knowledge discovery. Ontologies are meant to define a domain and to be shared & used by many, therefore most successful ontologies were built by expert groups and also received the domain community acceptance, i.e.: EngMath in Engineering Math [1], OntoNova in Chemistry [2], GeneOontology in biology [3], etc.

Building an ontology requires first to provide a way to find and organize optimization-related information, knowledge, artifacts, and also to establish common vocabularies, nomenclatures and taxonomies in the field. Using the ontology helps increase interoperability, integration and reuse of optimization artifacts like algorithms, models, libraries or simulation tools.

Some attempts to create optimization ontologies can be found in the literature. These ontologies usually focus on the problem field (design, simulation, modeling) and they include an optimization part for the solution. SoPT & ONTOP are the ones closest to the current work. ONTOP (Ontology for Optimization), was developed to facilitate Engineering Design Optimization (EDO). The preliminary work began with the development of a Finite Element Model (FEM) knowledge-capturing tool, ON-TEAM [4]. ONTOP's structure provides the means to identify feasible optimization techniques, for a given design optimization problem. SoPT, the ontology for Simulation OPTimization includes concepts from both conventional/mathematical programming and simulation optimization. SoPT aims to describe simulation optimization methods and help detect the correct tool for each specific case and to facilitate component reuse, especially in systems where simulators and optimizers are loosely-coupled [5]. DeMO is an ontology for Discrete-Event Modeling (DEM) [6]. The DeMO ontology is aiming to the modeling and simulation field but it relates also to optimization as these three fields are highly connected and they complement each other. The main problem of most other ontologies is that they are structured around the domain of the problem to solve (design, simulation, biology, etc.) and less around the optimization domain. SoPT is attempting to correct it and the designers consider relating SoPT to the ONTOP & DeMO ontologies, but no .owl file is yet given for SoPT.

The present work investigates the development of an ontology for optimization that will be designed and structured with focus on the optimization domain and will be also capable to support any other application

domain with a minimal expansion. To create such flexible optimization ontology, it is necessary to decompose it in two parts, the core or common part of the optimization ontology and the domain or application specific part of the ontology. In this work the development of a main ontology is investigated first, and then as a case study, an .owl-based prototype example for workflow optimization is being discussed. The final goal of the workflow optimization ontology will be to provide a formalization of a generic design improvement cycle so that the iterative nature of the product design process can be effectively captured and described for both human and simulation workflows [7].

## 2 DESIGN OF THE OPTIMIZATION ONTOLOGY

The core ontology should include the definitions for typical optimization problems along with the descriptions of the methods and algorithms applied to solve an optimization task. The ontology's basic structure must support optimization processes in general and focus on how to select and apply a suitable solution for the optimization problem encountered. The ontology classes should eventually cover all entities that concur in an optimization task.

There are several parts that make up an optimization task. The main components are:

- The Optimization Problem Model,
- The Optimization Solution Method,
- The Optimization Algorithm

where, an optimization Problem Model will be solved by a Solution Methods that invokes one or more optimization Algorithms (figure 1).


Figure 1. The three main Classes

### 2.1 The problem model

The Model of each Problem should belong to one of the numerous optimization problem categories developed and studied during the last century, from linear programming to non-linear & stochastic, from local to global, etc. A detailed taxonomy of these problem model categories must be developed in order to correctly classify the problems posed (figure 2).

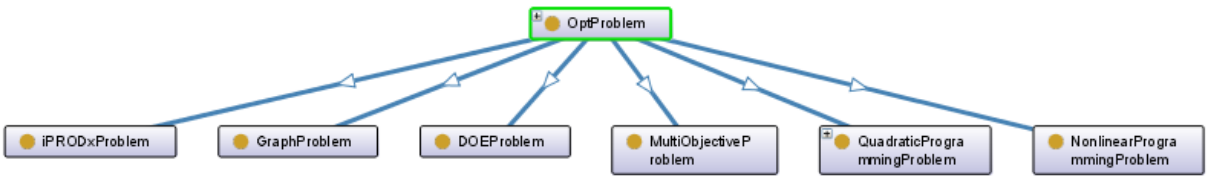


Figure 2. Problem Models

### 2.2 The solution method

For each category of optimization problems, at least one method has been developed to propose a solution. A Solution Method may solve several problem categories and a problem may be solved by several methods. Again a taxonomy of available solution methods must be developed in order to classify the numerous developed techniques (figure 3).

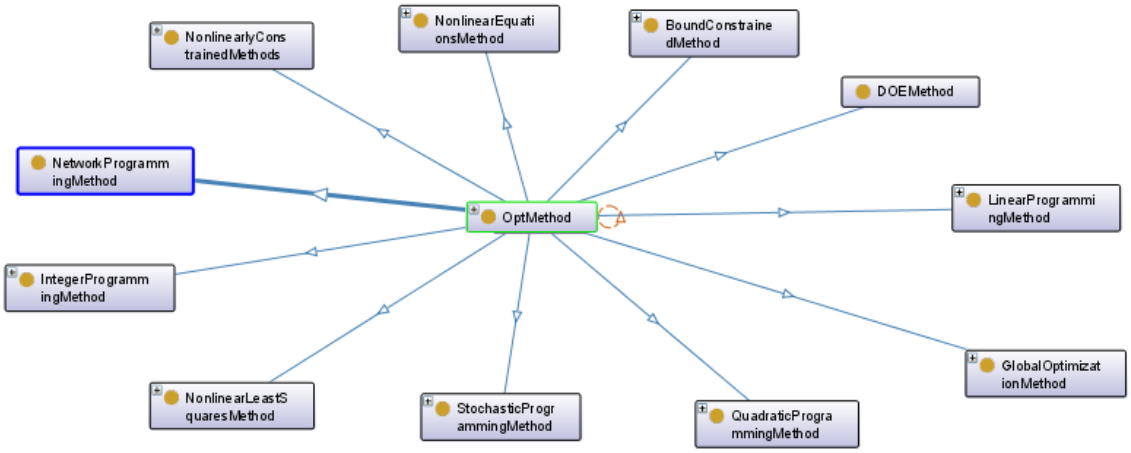


Figure 3. Solution Methods

### 2.3 The optimization algorithms

Usually the Solution Method & the Optimization Algorithm are seen as one thing. Optimization solutions are designed around an optimization algorithm. But, some methods may use more than one algorithms, or choose between versions of an algorithm or combine an optimization algorithm with other techniques. Therefore it is more flexible to separate the methods from the invoked algorithms, as they are designed to solve more basic tasks and they are implemented in specific modules, tools or applications. A taxonomy of all available algorithms must be developed to support the Solution Methods (figure 4).

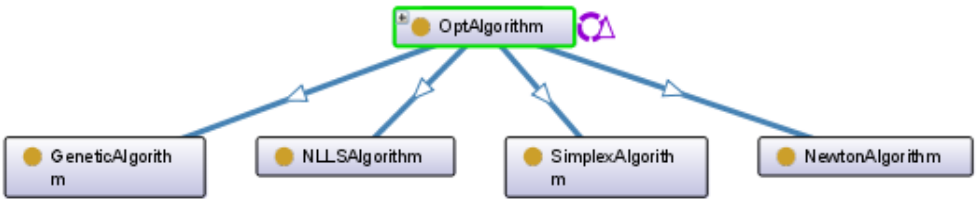


Figure 4. Optimization Algorithms

Problem models, solution Methods & optimization Algorithms are characterized by a number of Characteristics which relate to the type of the optimization problem under question. They are also accompanied by a number of Components required to pose and solve each problem correctly.

### 2.4 Optimization components

The basic components of any optimization problem are the Constraints & the Objectives, for posing the problem, and the Input/Output data & the parameters, for solving it. A collection of the various components that complement an optimization task should be put in the ontology to ensure the complete representation of problems and solutions (figure 5).

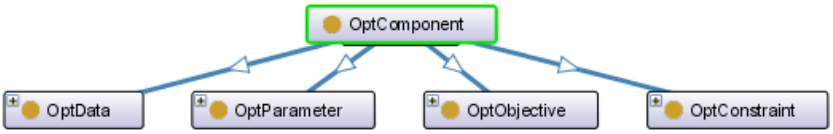


Figure 5. Components of an optimization problem or method

### 2.5 Characteristics of the entities

All three entities share some common Characteristics that categorize them and help us detect the appropriate algorithm and method to solve each problem. These characteristics, either general or specific, should satisfy certain rules in order to connect a specific problem with a specific solution (figure 6).

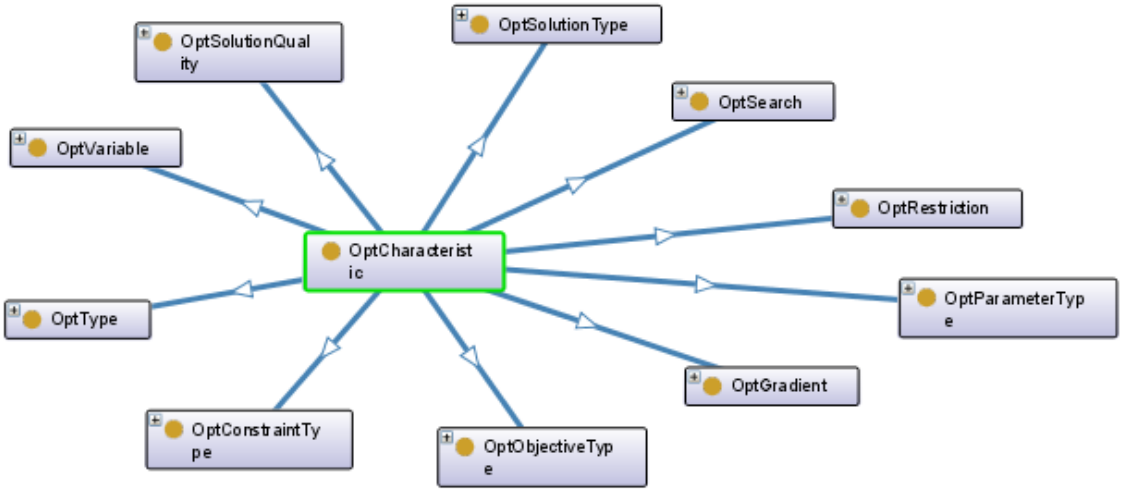


Figure 6. Various Characteristics of the entities used

Having built the above structure, the ontology under development will be able to: a) suggest the suitable solution methods & algorithms for an optimization problem (based on its characteristics) and b) check if the problem is accompanied by all information required by the method to run (based on required vs. available components).

### 2.6 Specific Entities

Further development of the ontology can be done by adding application, domain, or, company specific information, such as: available SW tools to run an optimization method, HW infrastructure supporting SW tools, processing jobs categories, etc. Having this information available, more precise suggestions are possible e.g.: What SW to use, on which computer and how to run the job.

### 2.7 Properties

Properties connect all the above classes and build their relations. For example, the *hasCharacteristic* property enables us to identify compatible methods & algorithms with problems as they should possess the same characteristics, equally, the *hasComponent* property enables us to detect the required components to process a solution.

### 2.8 Rules

Based on the proposed optimization ontology several useful semantic rules on e.g. algorithm selection can be derived (e.g. if *ObjectiveFunction* is X and *OptSolutionQuality* is Y then *OptAlgorithm* is Z).

## 3 BUILDING THE ONTOLOGY

The ontology was built using the Protégé-OWL tool and editor [8] shown in figure 7.

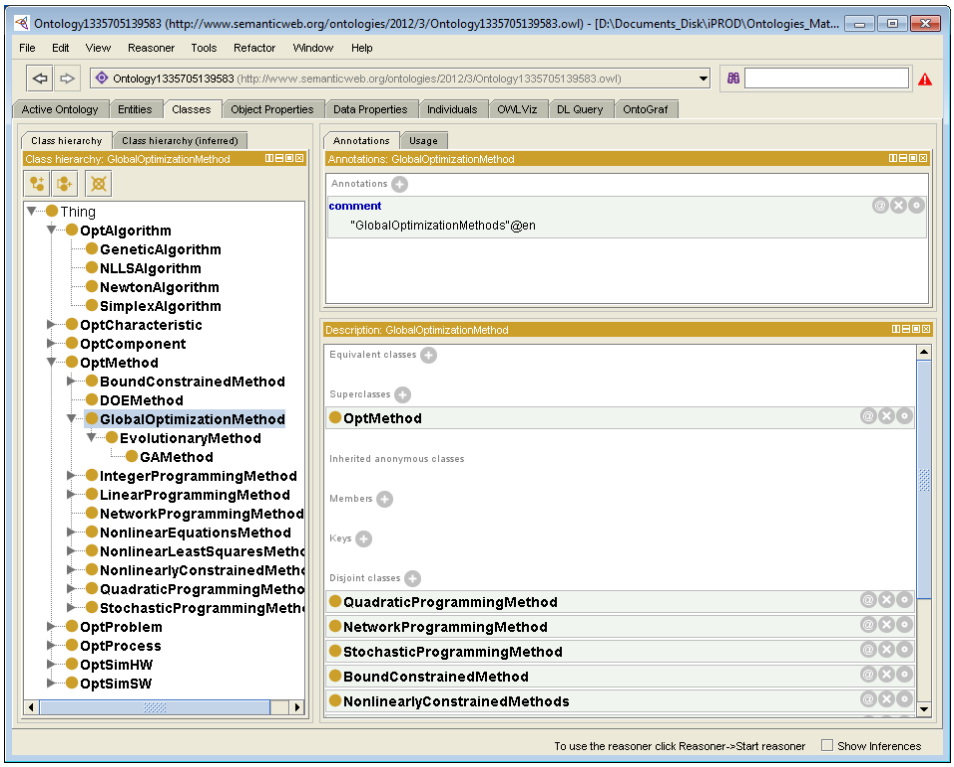


Figure 7. The Protégé tool interface

A snapshot of the classes and properties introduces is shown in figure 8. Each subclass has a tree-style structure. There is no limit to expand the Problem, Method and Algorithm classes contained in the ontology, as there are hundreds of variations or sub-categories.

The relationships between classes, subclasses and also their properties are shown in a network graph in figure 9.

The development of an ontology is a never ending process that continuously enhances and updates the classes, the properties and the instances in the ontology.

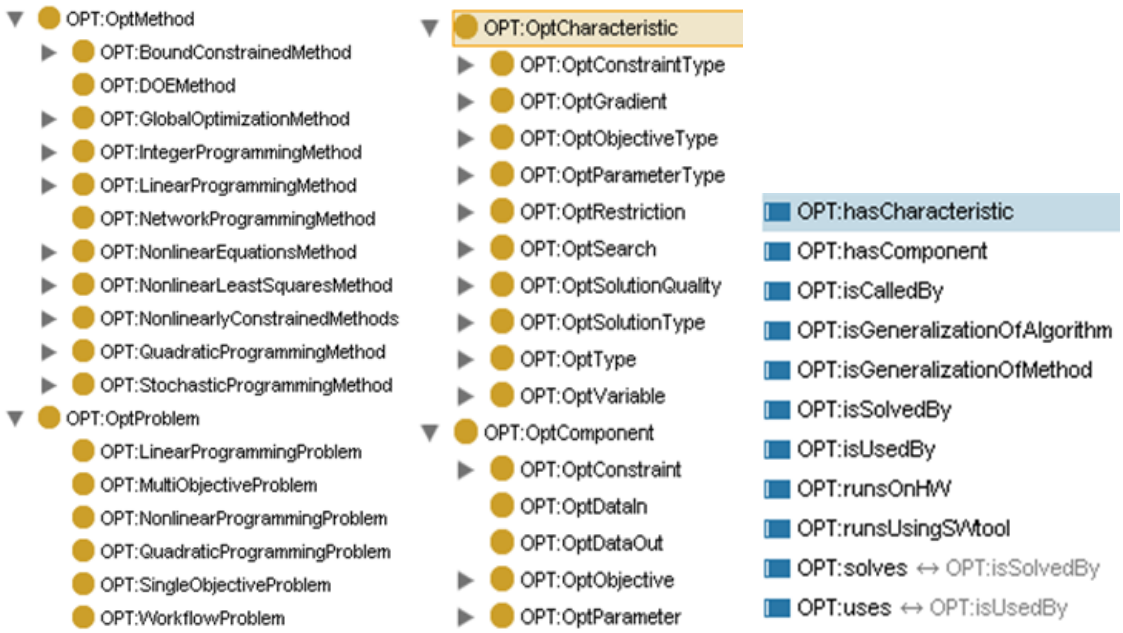


Figure 8: Optimization Ontology: Sample lists of Classes/Subclasses and Properties.

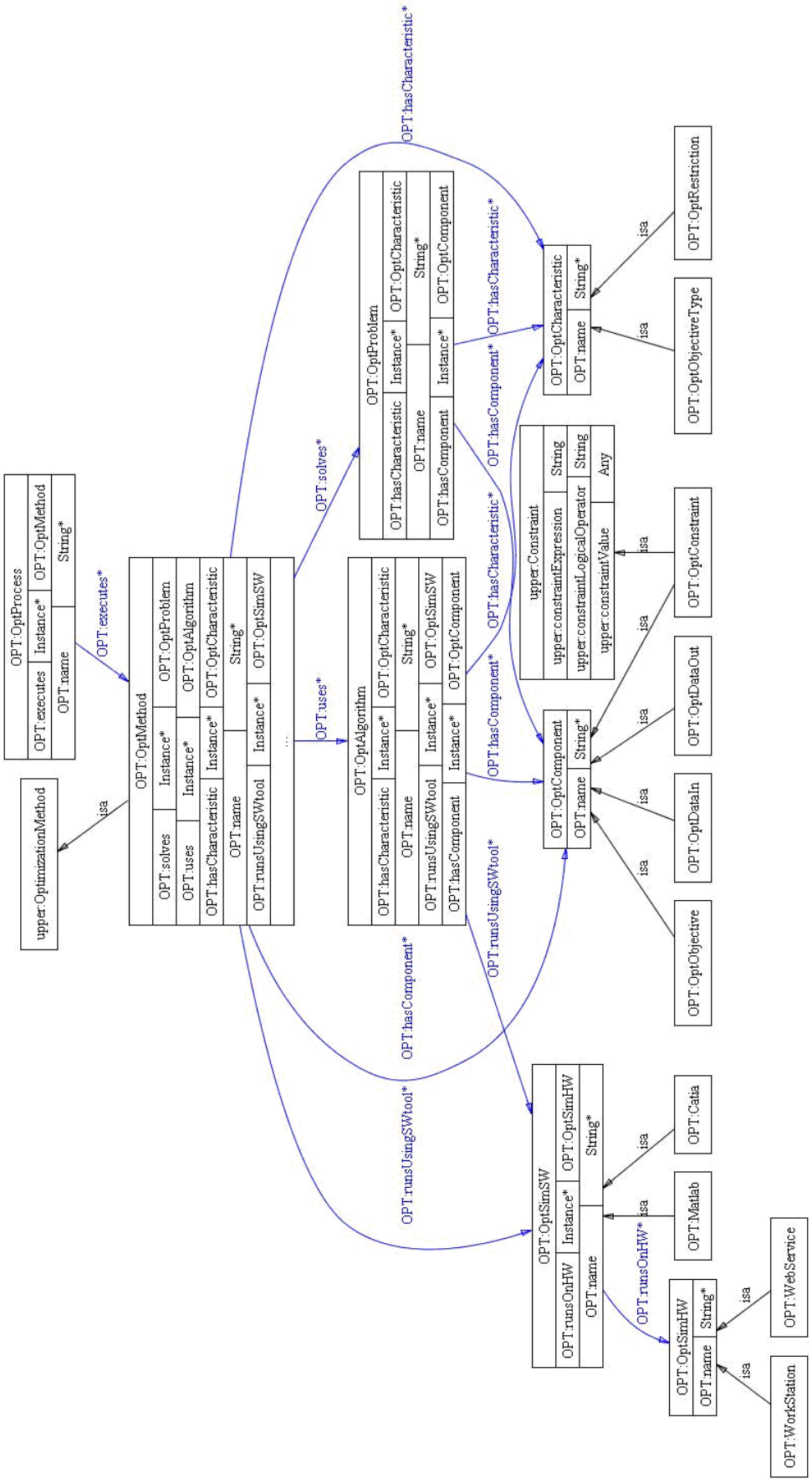


Figure 9. Classes and relations visualization of optimization ontology

## 4 CONCLUSIONS

The aim of this paper was to investigate the development of an optimization ontology that can support automatic selection of the appropriate optimization tool for a given problem. The authors developed the basic concepts common to all optimization problems that are required by the core part of the ontology, and defined how that could be complemented by domain specific modules tailored to the applications/problems under investigation. Future work includes the application of the ontology on simulation workflow optimization problems running in heterogeneous environments.

## 5 ACKNOWLEDGMENTS

The work presented in this paper has been partially funded by the European Commission and was performed under the framework of the FP7 ICT project "Integrated Management of Product Heterogeneous Data" (iProd), contract number FP7-ICT-2009-5-257657.

## REFERENCES

[1] Gruber Thomas R. and Olsen Gregory R. (1994), "An Ontology for Engineering Mathematics", *Fourth International Conference on Principles of Knowledge Representation and Reasoning, Gustav Stresemann Institut, Bonn, Germany, Morgan Kaufmann, 1994.* http://www-ksl.stanford.edu/knowledge-sharing/papers/engmath.html.

[2] Angele Jürgen, Moench Eddie, Oppermann Henrik, Staab Steffen, Wenke D. (2003), "Ontology-Based Query and Answering in Chemistry: OntoNova Project Halo", *Lecture Notes in Computer Science Volume 2870, 2003*, pp 913-928.

[3] The Gene Ontology Consortium. Gene ontology: tool for the unification of biology. Nat. Genet.. May 2000;25(1):25-9, http://www.geneontology.org/.

[4] Witherell, P., S. Krishnamurty, and I. R. Grosse. (2007). "Ontologies for Supporting Engineering Design Optimization". Journal of Computing and Information Science in Engineering 7(2):141–150. http://www.center4edesign.org/index.php/about/completed-projects-category-list/156-ontology-to-support-engineering-design-optimization.

[5] Han Jun, Miller John A., Silver Gregory A., (2011) "SOPT: Ontology For Simulation Optimization For Scientific Experiments", *Proceedings of the 2011 Winter Simulation Conference, IEEE* p.2914, http://www.informs-sim.org/wsc11papers/260.pdf.

[6] Miller, J. A., G. T. Baramidze, A. P. Sheth, and P. A. Fishwick. (2004). "Investigating Ontologies for Simulation Modeling". *In Proceedings of the 37th Annual Simulation Symposium, edited by H. Karatza, ANSS '04, 55–63. Washington, DC, USA: IEEE Computer Society*. http://www.cs.uga.edu/~jam/jsim/DeMO/.

[7] iProd project, Integrated management of Product heterogeneous data, http://www.iprod-project.eu/

[8] The Protégé Ontology Editor and Knowledge Acquisition System, (2013) http://protege.stanford.edu/