17th European Symposium on Computer Aided Process Engineering – ESCAPE17
V. Plesu and P.S. Agachi (Editors)

A Web-based Infrastructure for Integrated Life-cycle Engineering

Rafael Batres, Kazumasa Hayashi, Yoshiaki Shimizu

Toyohashi University of Technology, Hibarigaoka 1-1, Tempaku-cho, Toyohashi 441-8580, Japan, E-mail: {rbp, khayashi, shimizu}@pse.tut.ac.jp

Abstract

This paper presents the development up to date of a computer framework for integrating LCE methods and tools aimed at supporting ecological and economical decision making of product chains in a collaborative way. In particular, this paper proposes an approach to assist engineers and other decision makers in finding models that can be assembled into combined process that satisfy the product requirements.

Keywords Life Cycle Engineering, model composition, web-based decision making, knowledge integration

1. Introduction

Life-cycle engineering (LCE) is a set of engineering activities that include the application of engineering and scientific methods and tools to the design and manufacture of products, characterized as activities that are environmentally and economically efficient [1]. Computer-aided applications have been developed that support a number of LCE methods and tools. However, many of these applications operate independently with a limited ability to carry out decision making on the whole product chain.

This paper presents the development up to date of a computer framework for integrating LCE methods and tools aimed at supporting ecological and economical decision making of product chains in a collaborative way. Life-cycle inventory analysis tools are used to quantify the usage of raw materials, energy and generated emissions, waste and products of each individual process including material and energy flows that are delivered to other processes. These and the rest of the applications of the framework are distributed on the network and can be deployed through the Internet. Accessibility from the Web allows people that are geographical distributed to reuse and share existing models and tools.

2. Methodology

An LCE model is built to analyze environmental, societal, and economic aspects of the product system or product life-cycle. In the proposed framework, LCE models can be created from scratch or by combining models of individual components such as models of factories. Also, users can search models that satisfy certain criteria by means of matchmaking their capabilities and constraints.

Each model is managed by a model profile agent (MPA) which is a service provider of modeling capabilities based on execution platform, process, and model inputs and outputs. In the model profile (Figure 2), a model has two kinds of inputs and outputs. Firstly, there are inputs and outputs that refer to physical quantities that are associated to variables in the model. Secondly, there are inputs and outputs that refer to the computation that the model provides. Physical inputs and outputs are managed by the process. The process object represents an actual industrial process in the LCE system that accepts feedstock material that is transformed into other substances and energy. The MPAs advertise services with the Directory Facilitator who manages the yellow pages of all the applications of the framework. MPAs accept messages from model service requesters to obtain the degree of matching between the service requirements and the capabilities of the model known by the MPAs.

Information about the processes, physical objects represented in the model, and physical quantities is represented using classes and relationships defined shared ontologies implemented in the OWL language [2]. Ontologies define classes and properties of things such as substances, and processes using a modeling approach which is application-independent. The constraints of the models are encoded in CLIF (Common Logic Interchange Format) which is a simplified and updated version of the knowledge representation language known as KIF (Knowledge Interchange Format). Evaluation of the constraints is done by means of inferences made by a set of first-order logic reasoners. The actual content of the model is encoded in a predefined XML format. For example, the constraint below specifies that the model offered by the MPA is a model of an estherification process in which there is a variable for calculating the flow of carbon dioxide.

```
<must_satisfy_constraint>
```

```
  <constraint_weight>55</constraint_weight>
  <constraint_code>
  (and
   (lce:model_of_process ?model ?process)
   (rdf:type ?process lce:esterification)
   (lce:has_consumable ?process ?inputstream)
   (lce:has_product ?outputstream ?output)
   (rdf:type ?output lce:biodiesel)
   (lce:has_product ?outputstream ?output)
   (rdf:type ?quantity pq:physical_quantity_of_individual)
   (holds ?relation ?co2 ?quantity)
   (rdf:type ?co2 lce:carbon_dioxide)
   (rdf:type ?quantity lce:kilogram_per_hour) …)
   </constraint_code>
</must_satisfy_constraint>
```

In this example, *?relation* refers to a relation between the model and the quantity named and defined by the model developer. It is not necessary to know the name of that relation and instead use the fact that such relation is an instance of **class_of_indirect_property**. The **class_of_indirect_property** is a **rdfs:subClassOf owl:FunctionalProperty** whose domain is given by members of class_of_individual and whose range is given by members of **property_space**. Classes like **class_of_indirect_property** are defined in the upper ontology and therefore are common and shared between the agents. The upper ontology is based on [3].

3. Agent architecture

The agents of the framework are composed of four components: a communication component, a reasoning component, a control component and a graphic user interface. The communication component describes the message templates and the actions to be taken when a message is received. The

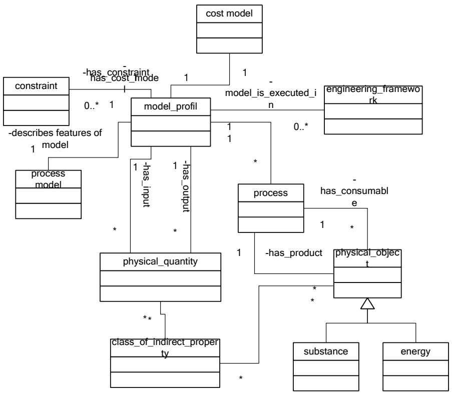

Figure 1. Model profile and its relations

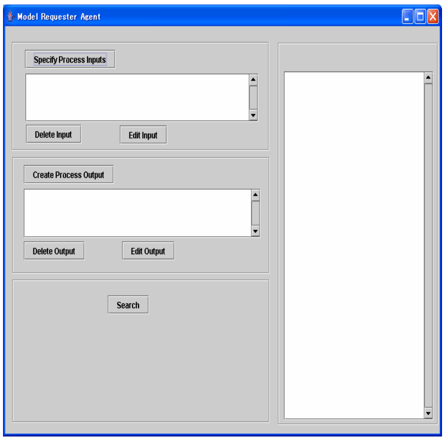

Figure 2. Screendump of the MRA

reasoning component manages the ontologies for knowledge processing. The graphic user interface of the Model Requester Agent (MRA) is used to specify queries and to visualize the results of the matchmaking.

4. System implementation

The prototype was programmed in Java using the JADE (Java Agent DEvelopment Framework) library for distributed agent applications and the JTP inference system [4].

JADE provides a Java library that can be used to implement multi-agent systems. JADE uses a middle-ware that complies with the FIPA specifications. The agent platform can be distributed across machines. It also provides tools for monitoring and configuration.

In the JADE runtime environment, agent communication follows the FIPA standard, which is maintained by the Foundation of Intelligent Physical Agents [5]. The Directory Facilitator (DF) is a component of the Agent Management Reference Model described in the Agent Management Specification.

Messages are encoded in FIPA ACL (Agent Communication Language). A message contains a number of parameters such as performative, sender, receiver, content, language and ontology. The performative defines the declarative act. The matchmaking environment implements the *request*, *query-ref* and *inform* performatives.

MPAs advertise its services with the DF by sending a *fipa-request* message with the registration request in the content of the message. Also a MRA can make use of the yellow page services of a DF by sending a *fipa-request* message. After getting the list of all available MPAs, a MRA prepares a list of feedstck requirements and product specifications and submit this information to MPAs by means of a *fipa-query-ref* message. Each MPA then sends a numeric score that represents the degree of matching. Similar communication acts are used for obtaining the classes of subprocesses used in by the process in the profile of the MPA.

Queries to the ontology are passed to JTP (Java Theorem Prover) which is a reasoning system that can derive inferences from knowledge encoded in the OWL language. JTP is composed of a number of reasoners that implement algorithms such as generalized modus ponens, backward-chaining, and forward chaining and unification [6] JTP translates each OWL statement into a CLIF sentence of the form *(PropertyValue Value Predicate Subject Object)*. Then it simplifies those CLIF sentences using a series of axioms that define OWL semantics. OWL statements are finally converted to the form (Predicate Subject Object). Queries are formulated in CLIF, where variables are preceded by a question mark. Following are some of the queries.

All agents have copies of the upper and domain ontologies that can be retrieved from an Internet agent. MRA use JTP in a number of ways. For example, when the decision maker defines a search session, the MRA list the classes of

processes. This list of classes is obtained by querying the JTP knowledge base with the following query: `(rdfs:subClassOf ?x lce:process)`
This means that agent code remains unchanged even when new classes are added to the ontology file. JTP is also used dynamically present user interfaces based on the information selected by the user. For example, if the decision maker is to define the **water_content** of a feedstock the MRA presents a screen for entering the value and unit of measure associated to the **mass_quantity**. However, if the decision maker defines the phase of the feedstock then the MRA presents a screen for specifying whether it is solid, liquid or gas. Again, there is no need to modify the agent's code if new units of measure or new properties are added to the ontology.

The inference engine is also used for determining the degree of matching between the requirements (feedstock and product specifications) and the constraints of the process. When a MRA ends a message to a MPA asking for the degree of matching, the MPA asserts the feedstock and product information as facts in the knowledge base. This is possible because MRAs send requirements the form of fictitious instances of models, processes, etc. Subsequently, the MRA checks the requirements against the constraints and the rest of the information of the model profile. As mentioned earlier, process constraints are in fact queries to the newly asserted instances. A matching score is assigned depending on the weight of individual constraints and the number of positive answers to the queries.

Queries are represented in OWL-QL which is a formal language and protocol for query-answering dialogues among Semantic Web computational agents [1]:

```
Query:  ("Check this constraint")
Must-Bind Variables List:  (?process)
May-Bind Variables List:  ()
Don't-Bind Variables List:  ()
Premise: process (requirements submitted by a PR)
Query-pattern: constraint in CLIF
...
```

5. Case study

A life-cycle model is to be built out of model elements. The objective is to evaluate the cost, societal and environmental implications of biodiesel obtained out of a specific kind of grain. The decision maker will evaluate cost, energy consumption, CO2 emissions. Consequently, the requirements are specified as ontology instances. The overall instance is an instance of an unknown model that will be used to calculate cost, energy, and CO2 emissions which are physical quantities associated to the biodiesel production process. The first step is to identify those models that meet these requirements.[7] Therefore, the MRA sends this model instance to the MPAs who are registered with the directory facilitator of the agent architecture. In this case, model profile agents MPA1 and MPA4 report the highest score with 100% matching the feedstock, product and model output requirements. MPA2 and MPA3 show only 50% of matching as they represent models that can convert the grain and calculate the required cost and environmental parameters. However, MPA2 and MPA3 produce fuel gas but no biodiesel.

6. Conclusions

This paper discussed the matchmaking component of a LCE framework which identifies potential models that match a set of requirements for assembling models of product systems and evaluate them. The proposed approach is to deploy cognitive agents that either advertises or requests information on process technologies. For matches with low scores, ongoing work includes the development of brokering algorithms that subcontract services to other agents so as to satisfy constraints.

Acknowledgements

Authors acknowledge the support of the foundation for the Fusion of Science and Technology (FOST).

References

1. M. Hauschild, J. Jeswiet, L. Alting. Anals CIRP, 54, 2 (2005) pp. 535-555
2. W3C, 2004, OWL Web Ontology Language Overview, W3C Recommendation, [Online] Available: http://www.w3.org/TR/owl-features/
3. ISO 15926-2, 2003. ISO-15926:2003 Integration of lifecycle data for process plant including oil and gas production facilities: Part 2 – Data model
4. R. Fikes, J. Jenkins, and F. Gleb. JTP: A System Architecture and Component Library for Hybrid Reasoning (2003)
5. FIPA (2005). FIPA specifications. [On-line] Available: http://www.fipa.org/specifications/index.html
6. S.J. Russell, and P. Norving. Artificial Intelligence: A modern Approach, Englewood Cliffs, NJ, USA, Prentice Hall (1995)
7. Fikes, R., P. Hayes, I. Horrocks (2003). OWL-QL A Language for eductive Query Answering on the Semantic Web. [On-line] Available: http://ksl.stanford.edu/projects/owl-ql/