ELSEVIER

Journal of Materials Processing Technology 107 (2000) 187–193

Journal of Materials Processing Technology

www.elsevier.com/locate/jmatprotec

Corporate ontologies and concurrent engineering

Christophe Roche*

LGIS-LLP/CESALP, University of Savoie, Campus Scientifique, 73 376 Le Bourget du Lac Cedex, France

Abstract

Concurrent engineering is based on the co-operation and collaboration of multi-disciplinary people who need to communicate and exchange information. Communication, as well as knowledge sharing and exchanging, are the corner stones of concurrent engineering. But each enterprise's actor speaks his own language, with his own terms and meanings: the enterprise, and especially the virtual enterprise, is a Tower of Babel. Although the communication problem can be reduced, from a syntactic point of view, by using a single communication language, the semantics problem remains to be addressed. It means that two entities can communicate only if they agree upon on the meaning of the terms they use. Ontology, understood as an agreed vocabulary of common terms and meanings shared by a group of people, is a solution to that problem.

This article will present our approach illustrated by the Ontological Knowledge Station (OK Station®). The OK Station is a software environment for defining and using ontologies. It relies on sound principles taking into account epistemological and linguistic notions. The semantics of term is based on elementary units of meaning and binary concept trees. Due to such features, obtaining real agreement about the meaning of terms is a reasonable and reachable goal.

Keywords: Concurrent engineering; Communication; Knowledge sharing; Semantics; Term meaning; Terminology; Ontology; OK model; OK station

1. Introduction

Concurrent engineering relies on the co-operation and collaboration of multi-disciplinary people. Let us take the example of product design and manufacturing (see Fig. 1). Collaborators, from design to manufacturing, have to share and exchange information: client requirements, business information, simulation results, workshop loads, supplier delays, etc. as they have to co-ordinate their decisions in order to react as quickly as possible to changes. Concurrent engineering is a knowledge- and communication-intensive process [1,2].

2. The communication problem

Communication between people, organisations and software systems is difficult due to the fact that each of these actors speaks a different language. To address this problem, we need a common communication language that agents can read and understand. Using a single, normalised language like KQML [3] can reduce the gap of misunderstanding. Although there is no syntactic problem left and although such languages give some useful indications about the pragmatic content of the message (by using predefined performatives, i.e. commitment actions), the semantic problem has still to be addressed.

3. The knowledge sharing problem

We may note that no communication is possible — not to mention collaboration or co-operation — without a single understanding or at least compatible understandings of the meaning of the terms exchanged between the different actors. In fact, communication cannot be reduced to only exchanging data but must take into account the exchange of knowledge. Co-workers in an enterprise work at their own levels, using their own knowledge and engineering models. Furthermore, the software tools extensively used in concurrent engineering, requiring specific and dedicated representations, are more concurrent than collaborative. The way to address this problem is to define a shared understanding. Agreement must be achieved about the shared knowledge used as a communication medium among people and software tools. If communication is the first cornerstone to collaborative product development, knowledge sharing between actors is the second one.

* Tel.: +33-4-79-75-87-79.
E-mail address: christophe.roche@univ-savoie.fr (C. Roche).

0924-0136/00/$ – see front matter
PII: S0924-0136(00)00713-5

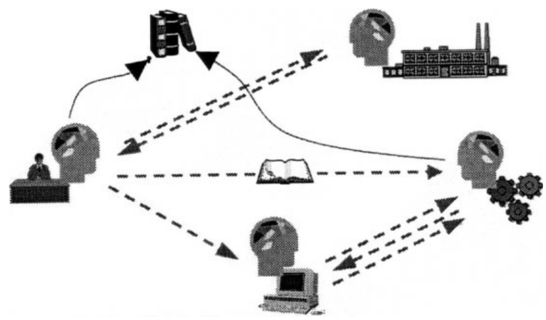

Fig. 1. A communicating enterprise.

4. Ontologies

Two entities can communicate only if they agree upon on the meaning of the terms they use. Ontology, understood as an agreed vocabulary of common terms and meanings within a group of people, is a solution to that problem.

It is amusing to remark that while the main goal of an ontology is to normalise, or at least to try to reach an agreement on the meaning of terms, the term "ontology" itself is not clearly defined and "seems to generate a lot of controversy in discussion about AI" [4,5]. When Gruber writes: "An ontology is a specification of a conceptualisation" [4], Guarino highlights the different interpretations covered by such a word: conceptualisation, formal ontology, ontological commitment, ontological engineering, ontological theory [5].

In the following, we will consider an ontology as a collection of agreements upon a vocabulary of common terms and meanings in some domains. We can identify three main categories of uses for ontologies in concurrent engineering, including enterprise modelling and multi-agent systems [6]. The first one is *communication* between people and organisations by providing a shared understanding. The second one concerns the *inter-operability* between systems where ontologies are used as interchange lingua. The last category is *systems engineering* by supporting the design and development of software systems: specification, re-usability and reliability. It is the reason why ontology can be considered as a key point for the next generation of concurrent engineering systems: SHADE for Shared Dependency Engineering [7], PACT for Palo Alto Collaborative Testbed [8]. Let us notice that concurrent engineering also relies on corporate ontologies which are a means to preserve corporate knowledge in order to better diffuse and reuse it.

Although ontologies have gained considerable popularity and some more or less general ontologies have been defined (Cyc, Mikrokosmos, Enterprise Ontology, TOVE, Sowa's, etc.) several problems remain which clamour for clarification. Although the problem of the different knowledge representation formalisms can be solved using an interchange format (like KIF for Knowledge Interchange Format [9]), the differences between the semantics of the systems, as well as not very clear epistemological principles, are barriers to the real use and re-usability of ontologies.

For example, how can we combine the definition of activity coming from the Enterprise Ontology TOVE [10]:

```
(define-class plan_action (?a) :def
  (forall (?alpha ?f ?s)
    (⇒ (holds (agent_constraint ?alpha (fluent_goal
    ?f)) ?s)
      (forall (?ap ?s1 ?s2)
        (⇒ (and (subaction ?ap ?a) (leq ?s1 ?s2) (Do
        ?ap ?s1 ?s2 (intended ?s2))
          (holds ?f ?s2)))))
```

with the definition of the same term coming from and Enterprise Ontology [11]?

```
(Define-Class Activity-Or-Spec (?X)
  "The union of Activity and Activity-Spec"
  :Iff-Def (And (Eo-Entity ?X) (Or (Activity ?X)
  (Activity-Spec ?X)))
  :Axiom-Def (Partition Activity-Or-Spec (Setof
  Activity Activity-Spec)))
```

Or how can we trust in, and then use, an ontology which does not offer "any kind of guarantees" [12]?

5. The OK model

The OK model relies on sound principles taking into account epistemological and linguistic notions: a concept is neither a class nor a set. The concepts, which represent the term's meanings, are structured into binary trees based on couples of opposite differences.

Our goal is to define the meaning of terms which refer to conceptual knowledge, i.e. abstract ideas like Human Being, Animal, whereas the word Peter refers to an individual.

5.1. Concept

There are two kinds of conceptual knowledge: concept and set. A concept is defined by the essence of the objects it subsumes and not by their state. Such a definition allows us to focus on the essence of the concepts and not on their state. An essence is invariant, which is not the case of state.

On the other hand, a set makes it possible to put together objects whose state shares some common properties. For instance, if 'Human Being' refers to a concept, 'Teenager' refers to a set composed of human beings whose age is under 18.

Let us note that the objects of a set can be of different natures. For example the 'Red' set gathers all the objects which contain an attribute called 'colour' whose value is 'red': red apples, red cars, and so on.

So, while an object is always subsumed by the same concepts, it can belong to different sets and not always the same sets according to its current state.