Knowledge Configuration in Virtual Enterprises

Nuno Silva and João Rocha
Departamento de Engenharia Informática
Instituto Superior de Engenharia do Porto
Instituto Politécnico do Porto
Rua Dr. António Bernardino de Almeida,
4200-072 Porto – Portugal
E-mail: {Nuno.Silva, Joao.Rocha}@dei.isep.ipp.pt
Web: www.dei.isep.ipp.pt/~nsilva

Abstract: With the expansion of the Internet, other networks and services, the society and its economic activities are changing drastically, either speeding the processes, exploiting word wide capacities, requesting products and services inaccessible in traditional context. Currently, both business organisations and private entities engage in electronic conversations and business activities using ad-hoc tools, like e-mail or web browsers. This is a user-based model since in most cases the user searches, mine and concludes the process with minimal help from the applications. On the counter part, the tool does not "understand" or even "remember" important user preferences or behaviours. From the business point of view, it is now consensual that organisations are evolving to more dynamic organisation forms like extended enterprise or virtual enterprises. Enterprises will focus its core competencies though outsourcing complementary products or even actively pursuit cooperation partners to mutually complement its activities. This paper characterises and describes concepts and requirements in Virtual Enterprises, E-Business and other complex interactions between intelligent information entities. Identifying such requirements, a set of functional services are proposed, which allows entities to adhere to a community and further accomplish common goals. The main efforts of the project are now focused in the knowledge level, especially in the automatic configuration of conversations and agreement on platforms for knowledge sharing between heterogeneous entities.

1. Introduction

Word wide economy dramatically changed over the last years, passing from a physical site based business, to a virtual site based business. The most paradigmatic example of such approach is the E-Business, which uses the Internet infrastructure to reach the most diverse costumers in the all-possible business fields. Assistants in shopping, travel, information retrieval, production planning and scheduling, etc., are already concrete examples how to exploit the new challenges and opportunities. Although these tools look promising, a new level of assistance must be achieved in order to change the user-based process to a machine-based process. The tools must encompass manners to increase the assistance process with user preferences and know how. On the inner side point of view, the traditional enterprise comprises most of the phases needed to provide a product or service from order booking through design, production, marketing, and lately, recycling and final elimination. The majority of these enterprises had the usual policy of directly controlling all the phases of business processes, within the enterprise using internal resources. However, different opinions (NGM, 1997), suggests that multiple trends are affecting business context, leading it to complete new challenges. In fact, the referred trends can be summarized in four distinct subjects: (i) Increased product variety over time, (ii) Increased technological complexity; (iii) Market globalisation; (iv) Increased social pressures (see Sousa *et al.*, 1999 for more extended considerations on reasons and implications of these trends). With the emergence of large, distributed, decentralised, flexible and very dynamic systems like the Internet, these properties become ordinary and its concrete value seems indubitable. Even if major progresses are crucial to accomplish many of the requirements already experimented, these concepts appear to be the correct approach. The question in most cases is the integration of disparate approaches, architectures, technology and above all, the knowledge management between different organisation domains.

The rest of the paper is divided in the following subjects. In Section 2, the most common virtual organisations characteristics and behaviours are described. In chapter 3, some of the identified operational services are described and its operation exemplified. In section 4, is presented an on-going project related to automatic integration of high-level communication contents and knowledge sharing. In section 5, a brief summary is reported and some further directions are presented.

2. Virtual Organisations

Enterprises are changing its structure and organisation, evolving to more dynamic organisation forms like extended enterprise or virtual enterprise (VE). Figure 1 (based in Park and Favrel, 1998) captures the organization evolution during the last decade throughout different challenges.

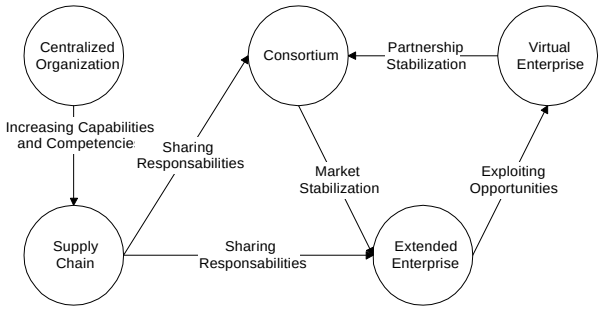

Figure 1 - Organization Evolution

Organization faces a constant and (almost) cyclic evolution process, adopting the most advantageous structure. After the supply chain stage, the organization adopts an extended enterprise or consortium structure. The organization can choose a mature market (Extended Enterprise) or choose to operate in dynamic market base (Consortium). The later stage of evolution is the Virtual Enterprise, characterized by its opportunistic nature. Although, this is not the final stage, as it is possible to evolve to market or partnership stabilization, continuing the evolution cycle.

Therefore, the Virtual Enterprise concept relates to a set of enterprises opportunely joined in a logical, momentary, although operational manufacturing structure, co-operating to accomplish predefined and mutually accepted goals. Such enterprises comprehend multiples autonomous entities, each potentially composed by other autonomous inter-related co-operative enterprises, behaving globally as single one. Each autonomous entity tend to focus its core competencies outsourcing complementary products or even actively pursuit cooperation partners to mutually complement activities (Camarinha-Matos and Afsarmanesh, 1997; Silva and Ramos, 1999), expecting to reduce risks and costs while maximizing market opportunities. This kind of enterprises “lives” in a very dynamic and challenging environment, where a new market opportunity, be a product or service, triggers the creation of a complete new logical enterprise structure providing the market with such new product. Deeper analysis remarks the need for important improvements in enterprises behaviours, namely complementing:

- Reactivity with Awareness and Partnership. The enterprise needs to be perfectly aware of market context and promote partnerships in order to opportunistically react to market needs and prevent its own limitations;
- Flexibility and Adaptability with Co-operation. In a highly changing environment, resources flexibility and adaptability is no longer sufficient to maintain enterprise expertise and competency. By its own, the enterprise tends to be insufficient to produce the entire final product. It must rely in others entities capabilities to ensure the execution of the rest of the product, i.e. co-operate;
- Autonomy with Agility. To focus and get advantage of its core competencies, enterprise must promote its autonomy and specificity. Considering the highly changing environment, enterprise requires to easily, rapidly and efficiently inter-relate with disparate enterprises, i.e. agility;
- Information with Knowledge. This is probably the major challenge the enterprise faces. Traditional enterprise considers its knowledge one of the most important competitive factors and it is not ready to share it with competitors. However, to know, to share and explore available knowledge can be a major advantage in order to achieve previous challanges.

Considering these premises, it is clear that the monolithic, centralised and static CIM approach do not fulfil modern enterprises functional requirements. Considering the manufacturing domain, several technological paradigms have been proposed, namely the Fractal Factory, the Bionic Manufacturing System, the Holonic Manufacturing System, and the Random Manufacturing Systems (Tharumarajah *et al.*, 1996). Although these are being adopted mainly in manufacturing systems, it is clear that many characteristics can be exploited to other domains or applications, without small inconvenience to the paradigm. In addition, combinations and interrelations of systems based in different paradigms are possible and beneficial to exploit the better performance of each other. A brief description of these paradigms can be found in (Tharumarajah *et al.*, 1996), while a perspective between these paradigms and the MAS technology adapted to manufacturing systems can be found in (Sousa *et al.*, 1999).

From the conceptual models mentioned above, through its complete realisation, a long-term research must be pursued in different research subjects, especially in organisational infrastructures and knowledge management.

3. Functional Services

Even though organisational infrastructures requirements highly rely on the employed organisational paradigm, various patterns were recognized, which allows systematising some common requirements, especially

concerning Entity, Group and Information (evolving to knowledge). Four distinct services are proposed: Identification, Domain, Pooling and Information. These services are provided by the InfSys system, which is composed by multiples InfSys entities. The InfSys system enforces operation rules to a community of heterogeneous entities and provides it with essential services. The rules and services are assured by the co-operation between the InfSys entities. Each of these complies with the InfSys system core principles (creation, evolution, coherence and high-level security), and implements different services according to its start up instructions.

3.1. The Identification Service

The Identification service concerns the inclusion into, and exclusion off the community. To adhere, each entity has to request it and complete the next three phases:

1. Identification. If the entity is not allowed in the system and it is not yet integrated, then it is excluded;
2. Authentication. This phase guarantee the entity is whom it claims to be. Currently it comprehends a bilateral authentication based on public/private key;
3. Registration. This phase correspond to the entity admission to the system. It corresponds to insert the entity's identification, location and so far pertinent information in the InfSys database. After this phase, the entity is a system full member, which means it can integrate any conversation or activity in the community.

When the entity leaves the InfSys community, the Unregistration phase is processed. This phase is the opposite phase of the three mentioned above. It occurs due to either unregistration request from the member itself or due to connection breakdown between InfSys and the member. When the unregistration occurs, the remaining members are informed about the fact and requested to discard any activity or information related to the unregistered entity.

3.2. The Domain Service

One of the most import characteristics observed in the recent organisations are the capacity to dynamically group in short-term organizations. A set of entities, dynamically or statically grouped together for a specific propose is named a Domain. A system is composed by several of these domains, forming complex hierarchical and heterarchical structures, depending on specific policies and goals. The fundamental behaviours of these organisations rely on the entity basic information about itself and the system, and the ability to dynamically participate in non-predefined structures and tasks. For example, in a fractal factory project each factory entity represents a potential project member (perhaps manager) that can be contacted at any time, to dynamically form its project team and perform the requested task. These entities do not know which project will come up or when, but they must be agile to perform it. At the begin each entity knows limited facts about the society it lives on. The entity does not know all the domains that (can) exist or it can participate, but should be supplied with enough information in order to dynamically set up structures/organisations that allows it to carry out its tasks. Thus, the primitive scenario of evolution of these structures can be described as a wider society (wider domain), composed by all the existent entities, allowing each one to virtually know and contact all the others. Thus, the domain concept represents an autonomous, self-regulated entity, with ordinary capabilities and constraints, specified either at developing time or as emergent structures. The internal organisation (i.e. hierarchical/hetararchical) should be defined at application level. The proposed Domain Service provides conveniences for domain management. Two type of entities are recognized, the creator/manager, and the entities who populate the domain. The service conveniences includes:

- Creation of new entity that will identify the domain, and represents the domain capabilities, behaviours and plans: the manager;
- Creation and Configuration of the InfSys domain service, as mentioned in section 3.1, and other required services;
- Contact and Request previously chosen entities, which may have no authentication possibilities (this topic will be further analysed).

The Domain Service is proposed as a mechanism to create a new instance of the entity with certain configuration arguments, namely identification, context and limited knowledge, conversations, features and roles. The status, error or result, resumes to the parent entity. The new entity maintains a connection with its parent, which allows it to connect to the rest of the world. Any relation with outside world should be requested to the parent. Unless requested, when the link is broken the new entity tries limitedly to contact its parent and then resigns itself and the respective domain.

As mentioned before, the entities have limited knowledge about the world, which limits the use of the identification and authentication mechanisms. The preferred solution is the primary entity to generate password to the entity and deliver it prior to the domain creation. Hence, it is possible to the new domain to contact and securely authenticates entities. Another solution is the synchronisation of addresses and identification between creator and the primitive copy of the entity to be incorporated in the domain. Thus, the entities are accepted into the domain even if no domain specific password is presented.

Additionally, it is fundamental the information sharing between the primitive entity and its instances. Excepting the well-known permissions mechanism (similar to that of O.S.) no other concurrent mechanisms were applied. The permissions are set at creation time but can evolve throughout the operation, depending on specific application. The InfSys allows sharing information within domain, and entities taking part of multiple domains are able to publish and exchange domain information coherently.

Concluding, this service allows logically grouping multiple entities that behave globally as hierarchical systems, proving and interfacing with its lateral partners. Each of these elements forms a completely independent information context imposing publishing restrictions. The mentioned conveniences, except those related to the manager specific role, are also observed in the populating entities conveniences side.

3.3. The Pooling Service

The Pooling Service concerns with message delivery to entities temporarily unreachable or unregistered in InfSys. The pooling entity stores and delivers the requested message as soon as the destination entity is reachable. Each entity providing the service has to register and further publish its capability in InfSys Information DB (section 3.4) as any other capability. In order to achieve reliability (even it is not frequently demanded), multiple pooling entities are allowed in the system. Concerning this service, no co-ordination or co-operation is required, because any message is stored in pooling entity DB without any processing. Pooling service entity should periodically query its Identification DB for the destination entities and when available, send it the message.

3.4. Information Services

The service intends to supply the community with domain public information, namely capabilities one wants advertise in order to be requested, e.g. performing operations, reliability and costs. In order to achieve coherence, reliability and efficiency purposes, multiples InfSys are proposed/recommended, though incoherence easily occurs in such conditions. Consequently, InfSys co-operation and publishing rules are essential. The proposed rules follow:

- Information has a single owner/publisher, who has to be a registered member;
- The owner is responsible for information management (publish, update and unpublish);
- Only InfSys provides published information to the community;
- Owner limits information time validity. Information can not be updated/unpublished until the supplied information looses validity unless all the intervenient accept the update;
- When queried about unfamiliar information, InfSys query its partners for it;
- Information is valid exclusively during owner InfSys registration.

Any registered entity publishes its information in its InfSys entity, which in turn provides it when requested to the domain, therefore acting as a broker. Guarantying entity unique connection to the community, information coherence and validity are easier to maintain. Concerning information management, two different procedures occur: publishing and brokering. When an entity publishes information, it structures and tags the data using the following InfSys syntax (in BNF):

```
<what_publish>::=[<facts>]
<facts>::=<fact>|<fact>,<facts>
<fact>::={<name>,<description>,<struct>}
<struct>::={<type>,<data>}|
           [<fact>,<facts>]
<name>::=term
<description>::=string
<type>::=int|number|term|list|tuple|string
<data>::=<int>|<number>|
          <term>|<list>|<tuple>|<string>
```

The concepts in `<data>` represents the physical value specified in the native language (e.g. a list in C is not specified as in Prolog). When requesting the information, the entity must provide information constraining both the form and the content (values). These constraints are specified using the following nomenclature (in BNF):

```
<what_search>::=[<facts>]
<facts>::=<fact>|<fact>,<facts>
<fact>::={<search>,<constraints>}
<search>::=<name>|<description>
<name>::=term
<description>::=string
<constraints>::=<constr>|[<fact>,<facts>]
<constr>::=<data>|[<guards>]
<data>::=<int>|<number>|
          <term>|<list>|<tuple>|<string>
<guards>::=<guard>|<guard>,<guards>
<guard>::={<operand>,<operator>,<operand>}
<operator>::=<|=|>|~=
<operand>::=<data>|<fact>
<data>::=<int>|<number>|
          <term>|<list>|<tuple>|<string>
```

The complete grammar is not presented to simplify and maintain it short, but the presented one is not complete. For example, the combination of operands and operator are not all possible. The `~=` (correspond to SQL Like operator) operator is possible only with string operands.

The Information service is currently being expanded in the project described in section 4.

3.5. Application example

Prior to any application examples, it worth mention that the InfSys system does not intends to be another architecture standard, but in the contrary, adopt as many

standards as possible. For example, KQML specification is used, whereas the FIPA ACL specification will be allowed in the near future. Although, the InfSys clients no need to manipulate these messages in the InfSys protocols, since the supplied API entirely encapsulates protocol complexity. The API is developed in Erlang, a functional declarative language developed by Ericsson Telecom AB.

Any of these services were developed as independent layers. Additionally, providing both server and client support, task specific entities are able to provide and require these services. Specific application behaviour is developed as additional layer. Figure 2 represents this approach.

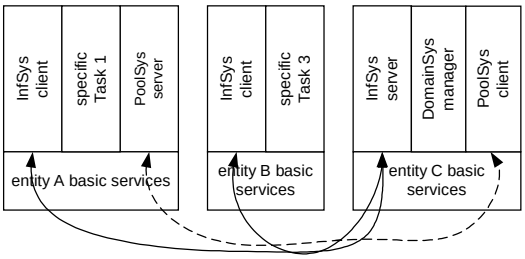

Figure 2 - Functional Services Implementation

Respecting the application development, both initialisation and further operation of these services are requested with simple functions calls. For example, to an entity start its InfSys server conveniences it calls the function:

```
infsys:start(Args)
```

where `Args` is Prolog-like list of tuples that optionally define the domain name to adhere, the broadcast port number to listen contact messages and various timeout values:

```
[{domain,planner},{domain_server,planner_1}]
```

From the client side, to adhere to a specific domain it only has to call the function:

```
infsys_client:connect(Args)
```

where `Args` optionally define, domain requested name, InfSys server name, broadcast port number and various timeout values.

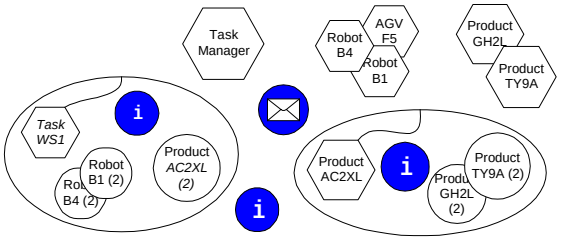

Figure 3 – Example of domain representation

The hexagonal forms represent primitive entities, and circle forms represents derivate entities. Notice that Product AC2XL is primitive entity composed by two derivate entities.

Imagine that, at some point in time a new order arrives to TM to produce *n* units of Product TY9A. A new task (project in the Fractal Factory concept) is triggered considering several constraints, which the most important (at this time) are the operations to perform and the required resources. The Task Manager prepares to create a new domain composed by a task entity that is composed by the application specific behaviour plus the conveniences supplied by the DomainSys package. The command function is:

```
domainsys:start(Args)
```

where `Args` is a Prolog-like list of tuples, that includes the new domain name, the manager entity function and any information concerned with the set-up of the domain, like number of InfSys and PoolSys servers to create, entities to allows entrance without registration, etc.:

```
[    {domain_name,"TASK TY9A-2"},
     {domain_manager,{taskmanager,task,[]},
     {domain_entities_allowed,["AGV F5"]},
     {domain_entities_mandatory,["AGV F5"]}
]
```

The `domain_manager` argument defines the manager execution module, function and its initial arguments. The `domain_entities_allowed` list, defines the entities which authentication based in password is not necessary. Finally, the `domain_entities_mandatory` argument specifies the entities which participation is mandatory.

4. Knowledge Sharing project

The Knowledge Sharing project is an on-going project intending to enhance the prior services concerned with the dynamic configuration of high-level conversations contents and knowledge sharing between heterogeneous entities. The dynamic nature of the virtual organisations requires an automatic configuration process that allows the exchange and common understanding of the syntax, semantics and utilisation of data and information, i.e. knowledge sharing. The Ontology concept along with Knowledge Engineering techniques is one of the most promising and active areas of ongoing research word wide. Ontology is the result of the knowledge acquisition, systematisation, representation and other knowledge engineering activities (Milton *et al.*, 1999). It defines specific domain entities, its properties, relationships, constraints and behaviours (Gruninger and Fox, 1995). Throughout this project, Ontology is a specification of a conceptualisation (Gruber, 1993) one has about a specific domain, thus necessarily restricted in applicability and the represented viewpoints of the reality. As an example, different ontologies of the same reality may use different terms to mean the exact same concept or use the exact same term to mean very different concepts (Schlenoff *et al.*, 1999). Thus, the use of ontologies does not eliminate the need for configuration and common agreement on knowledge.

Ontology concept only postpones the problem. During the last years appear a large number of ontologies regarding different domains, emphasising the incompatibilities and difficulties to integrate ontologies. A proposed solution is the translation between different ontologies, performed by specific translators. However, in most cases these translators are hand written, denoting a very important time discrepancy between the need and the availability of such translator. A service that would negotiate these issues without (or minimal) human intervention would be a huge improvement during inter-domain co-operation. Such service should negotiate conversation languages, formats and protocols, and specially, negotiate and automatically construct shared (accepted) ontology.

Our approach consists in developing a translator that automatically negotiates ontology translation procedures. This translator would apply different AI techniques like Reasoning and Learning, with other less AI dependents, like Negotiation strategies, Natural Language processing and Dictionary and Thesaurus features, etc.

Two independent modules compose the translator: the on-line translation module, responsible for message translation between two ontologies already mapped, i.e., ontologies which translation process is already defined, and the off-line translation module, responsible to carry out the negotiation to achieve the agreement concerning ontologies mapping procedure. Figure 4 represents the proposed approach.

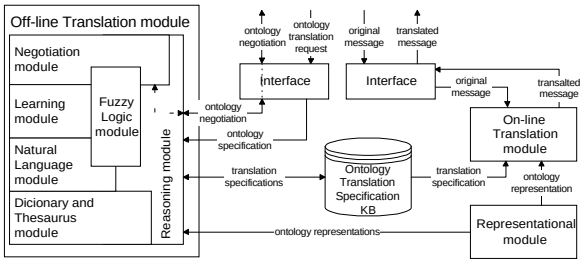

Figure 4 – Proposed approach schema

The Ontology Translation Specification KB is the repository of already achieved translation procedures. It serves to both on-line and off-line translation modules. However, its content is updated by the off-line module only when a translation agreement has been achieved between the module and the requesting entity. Within the off-line translation module, six different requirements are already identified, each implemented as a sub-module:

- The Natural Language (NL) module (also named Linguistic module) assists in translating ontologies described in terms of natural language. It is very important in communication translation based in natural language but less in data/information exchange, since its structure should be perfectly specified. For now, this module does not make part of our efforts, hence third part solutions have been used so far;
- The Learning module is responsible for the learning process, essentially related to patterns observed in ontologies translations;
- The Negotiation module is concerned with the negotiation with both the translation requester entity and the human expert. This module is of fundamental importance and is now the focus off our best efforts. It is intended to be an expert system, capable to combine different negotiation strategies accordingly to the negotiation context;
- The Fuzzy Logic module assists in proposing convergence approaches to negotiation, translation patterns and NL processing. This module, even if represented as a single module, it is composed by distinct sections each one related to its specialized applications;
- The Reasoning module is the off-line module core, requesting services from the prior modules and co-ordinating their activities. It is the responsible to process the ontology contents and based in its significance, define translation-mapping procedures. However, in most cases, the simple internal process is not sufficient to come out with a solution, are requiring negotiations and compromises (Fuzzy Logic module) between parts;
- The Dictionary and Thesaurus module assists the Reasoning and Natural Language modules in the syntax and semantics convergence. For the moment, third part solutions have been used.

Concerning the services described in section 3, they have been developed to incorporate de Ontology concept. This shifts the services to a higher knowledge level. Much ontology is being developed and is being used for many services and purposes, namely the information (knowledge) exchange. Each entity is allowed to structure its data/information according to an ontology, which allows the entities to reason on the data. The ontologies currently available are:

- InfSys ontology, that specifies terms and information structures related to the InfSys service. This ontology allows “legacy” systems to continue use its prior models;
- 2D Space ontology, that conceptualises the 2D physical space in a simplified form, specifying different scales and conversion routines;
- Time ontology, that formalizes time concepts, different scales and conversion routines;
- Personal Computer ontology, that relates several typical constituents of a personal computer, like the CPU, HDD, monitor, etc. This ontology had been used in case tests that explore the web for products.

At a future stage, it is reasonable to expect ontologies not only aggregate multiple others, but also rely in predefined and tested ontologies defined somewhere else. The most common ontology architectures will be composed by multiples layers, each inheriting and specializing multiple lower ones (Uschold *et al.*, 1998), creating even more detailed one (Figure 5).

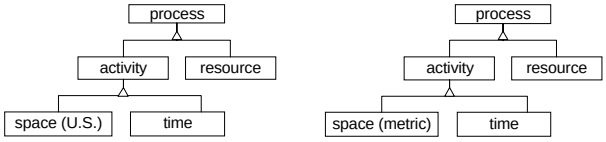

Figure 5 – Inheriting and specialising ontologies.

Therefore, the proposed strategy attempts to primarily combine lower layer ontologies, minimizing higher layer incompatibilities, creating KB ontologies translation specifications. These specifications will be then utilised not only to facilitate translation procedures but also in capturing translation patterns.

5. Conclusions

This paper discusses the future technological support of the Virtual Enterprise emergent organisational paradigm. Some common characteristics and behavioural patterns were identified which permit to suggest the relevance of unifying and ordering services. The proposed solution does not intend to be another standard architecture but considered as a valid approach and test case to some projects. The accumulated experience demonstrates that barely is found an optimal solution for all the situations occurring during project lifetime. Thus, we suggest the integration of different architectures and standards, which leads the project and any entity to easily incorporate heterogeneous communities, share capabilities and explorer remote resources. In this sense, many InfSys capabilities are possible since other resources are explored in other communities, like the Natural Language agent available in the Open Agent Architecture (OAA, 2000).

Considering this premise, it has been observed that it could be achieved only if knowledge management is contemplated, in opposition to the data/information management. Even if ontology concept seems to be the indubitable evolution vector in knowledge management, a step forward has to be done, incorporating ontologies interrelation, translation, aggregation, etc., in one word: plug-and-play. The proposed approach suggests the AI techniques and principles to achieve it. However, this is not an easy and rapid solution, since it relies in experts' knowledge. Also, and fundamentally, intensive experiments and respective feedback is time-consuming.

References

Camarinha-Matos and Afsarmanesh, 1997; Virtual Enterprises: Life cycle supporting tools and technologies; L.M. Camarinha-Matos, H. Afsarmanesh; in Handbook of Life Cycle Engineering: Concepts, Tools and Techniques, A. Molina, J. Sanchez, A. Kusiak (Eds.), Chapman and Hall, 1997.

Gruber, 1993; Toward principles for the design of ontologies used for knowledge sharing; Tom Gruber; in Formal Ontology in Conceptual Analysis and Knowledge Representation, edited by Nicola Guarino and Roberto Poli, Kluwer Academic Publishers; International Workshop on Formal Ontology; Padova, Italy; March 1993.

Gruninger and Fox, 1995; Methodology for the Design and Evaluation of Ontologies; Gruninger, M., and Fox, M.S. (1995); Workshop on Basic Ontological Issues in Knowledge Sharing, IJCAI-95, Montreal.

Milton *et al.*, 1999; Milton, N., Shadbolt, N., Cottam, H. & Hammersley, M.; Towards a Knowledge Technology for Knowledge Management; International Journal of Human-Computer Studies; September 1999.

NGM, 1997; Next-Generation Manufacturing; A Framework for Action; Executive Overview; January 1997; http://www.dp.doe.gov/ngm/ngm.pdf.

OAA, 2000; Open Agent Architecture, Developer's Guide (version 2.0); http://www.ai.sri.com/~oaa

Park and Favrel, 1998; Kyung Hye Park and Joël Favrel; Virtual Enterprise-Organization and Information Technology Infrastructure; 5th IFAC Workshop on Intelligent Manufacturing Systems (IMS'98); November 1998; Gramado, Brazil;

Schlenoff *et al.*, 1999; A robust process ontology for mfg sys integration; Craig Schlenoff, Rob Ivester, Amy Knutilla, National Institute of Standards and Technology Gaithersburg, MD 20899; URL: http://www.ontology.org/main/papers/psl.html.

Silva and Ramos, 1999; Proposal for Inter-Enterprises Negotiation Infra-Structures using an Holonic approach; Nuno Silva and Carlos Ramos; Proceedings of 1st International IFAC Workshop on Multi-Agent Systems in Production; Wien, Austria; December 2-4 1999.

Sousa *et al.*, 1999; Aspects of Co-operation in Distributed Manufacturing Systems; P. Sousa, N. Silva, T. Heikkila, M. Kollingbaum, P. Valckenaers; Proceedings of the 2nd International Workshop on Intelligent Manufacturing Systems (IMS-Europe'99); Leuven, Belgium; 22-24 September 1999.

Tharumarajah *et al.*, 1996; Comparison of the bionic, fractal and holonic manufacturing system concepts; A. Tharumarajah, A. J. Welles e L. Nemes; International Journal of Computer Integrated Manufacturing, 1996, vol. 9, nº 3, 217-226.

Uschold *et al.*, 1998; The Enterprise Ontology; Mike Uschold, Martin King, Stuart Moralee and Yannis Zorgios; The Knowledge Engineering Review, Vol. 13, Special Issue on Putting Ontologies to Use; Eds. Mike Uschold and Austin Tate; 1998.