

# Requirements for co-operation and knowledge sharing in Virtual Organizations

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:** Organisations in general and Manufacturing systems particular are changing its structure and behaviours, 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 activities. New organisational and technological paradigms had been proposed in order to reply to these challenges, but new technological challenges come out, especially in organisational infrastructures and knowledge management. This paper will characterise and describe concepts and requirements related to the most common approaches, emphasising its needs about organisational and technological shifts. Identifying such requirements, allows the specification and description of some basic functional services. Finally, it is presented an on-going project related to automatic configuration of high-level communication contents and knowledge sharing.

## 1. Introduction

The traditional enterprise comprises most of the phases needed to provide a product or service from order booking through design, production, and 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) Increasing social pressures (see Sousa *et al.*, 1999 for more extended considerations on reasons and implications of these trends). A simple observation concerning technology support shows the enterprise had been typically characterised by monolithic structure, centralised co-ordination, low flexibility and

adaptability, although highly efficient under intensive and repetitive processes (Sousa *et al.*, 1999). The Computer Integrated Manufacturing (CIM), the former major adopted technological concept, lacks the latest fundamental features such as distribution and decentralisation, flexibility and adaptability (Silva and Ramos, 1999). With the emergence of large, distributed, decentralised, flexible and very dynamic systems, which the Internet is the most paradigmatic one, 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 capacity to integrate disparate approaches, architectures, technology and above all, the understanding about each particular domain, application or simple terms used in transactions. New forms of inter-operability must be achieved prior to any new significant evolution in the distributed enterprise.

The rest of the paper is divided in the following subjects. In Section 2, the most common virtual organisations characteristics and behaviours are described, three organisational-technological paradigms are summarized, and its conceptual operation compared, which allows analysing requirement in operational services. In chapter 3, some of the identified operational services are described and its operation exemplified. In section 4, is presented an on-going approach related to automatic configuration of high-level communication contents and knowledge sharing. In section 5, a brief summary is done and some questions and potential directions concerning future work 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). VE is 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 (product for short), triggers the creation of a complete new logical enterprise structure providing the market with such new product. Deeper analysis of context changes, remarks the need for important improvements in enterprises behaviours, namely complementing:

- **Reactivity with Awareness and Partnership.** In order to opportunistically react to market needs, the enterprise needs to be perfectly aware of market context and promote partnerships in order to prevent its own limitations;
- **Flexibility and Adaptability with Co-operation.** In a highly changing environment, the enterprise 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. Thus, must rely in others capabilities to ensure the execution of part of the product (i.e. co-operate);
- **Autonomy with Agility.** The arising context do not represents a challenge to the enterprise autonomy. On the contrary, to focus and get advantage of its core competencies enterprise must promote its autonomy and specificity. Since the enterprise needs to co-operate and 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 largest challenge the enterprise faces. Traditional enterprise considers its knowledge one of the most important competitive factors. It really is, but in most cases, enterprise does not know, it simply process information or even just data. To know and to share such knowledge can a major advantage.

Considering these premises, it is clear that the monolithic, centralised and static CIM approach do not fulfil modern enterprises functional requirements. Hence, new organisational concepts must be analysed and validated. Three of these are briefly presented focusing its roots and fundamental features.

## 2.1. Fractal Factory

Fractal Factory paradigm is based on mathematical fractal concept and its associated theory of chaos. The fractal is characterised by constant evolution related to its partners and environment (Tharumarajah *et al.*, 1996) and internal operation, abandoning the function-oriented organisation and adopting a project-oriented organisation (Sihn, 1997). The organisational structure will encapsulate the process and the technology, forming a cybernetic structure composed by a more or less static set of resources and a very dynamic set of projects (tasks). As a new project is introduced into the system, each resource starts negotiation (it can even compete) with remainder resources, the execution of each sub-project. Thus, each project initiates a very dynamic process, responsible for resource/sub-project binding, leading to constant changes in the enterprise structure and organisation. This process is more dynamic (and presumable successful) as the resource can (well) decide about its own behaviour and does not rely on a higher entity to do it. Thus, the fractal factory must be understood as a non-linear system (Sihn, 1997), structurally reactive and adaptive to the dynamic context. In addition, the concept easily models different enterprise dimension, like strategic, social, cultural, informational and technological (Sihn, 1997).

## 2.2. Bionic Manufacturing System

Bionic Manufacturing System (BMS) is based on structures and behaviours observed in biology. Okino presented the Bionic Manufacturing Systems and the associated term *Biological-oriented*, which intends to apply to manufacturing systems, the structure and organisation behaviour of the living beings. From the simplest through the most complex living form, inside of certain hierarchically ordered relations, all manifest autonomy, spontaneous behaviour and social harmony. In biological systems, autonomy and spontaneous behaviour means each entity is responsible for its activities and self-division according to the genetic code stored in DNA<sup>1</sup>. Therefore, consistency and goal orientation are conceptually supported by the genetic inheritance. In order to co-ordinate and maintain the system vitality, the surrounding environment supplies both system memory and the co-ordinators (enzymes in biology). BMS tries to define a parallelism between biological systems and manufacturing systems, translating the DNA inheritance in biology to characteristics, states, behaviour and domain knowledge in manufacturing.

---

<sup>1</sup> The DNA is transmitted to descendants during the cell division. Each individual has its own DNA composition that makes it unique in the universe.

### 2.3. Holonic Manufacturing System

The Holonic paradigm arises from Herbert Simon and Arthur Koestler studies about biological society evolution and organisation. Simon observed that complex systems are hierarchical structured and formed by intermediate stable forms. These forms allow system to be stable, reliable and evolutionary, while maintaining a goal-oriented functionality due to its hierarchical structure. Later, analysing Simon theory and comparing it with its own observations, Koestler perceived that each system and its intermediate forms are simultaneously a part and a whole, a container and a contained, a controller and a controlled. In order to designate these hybrid nature and behavioural entities, Koestler (1967) proposed the term Holon as the combination of the Greek word *Holos* (whole), and the *on* suffix (particle, like in *neutron*). In addition, "Holarchy is a hierarchy of self-regulating holons, in supra-ordination to their parts, in sub-ordination to the higher levels and in co-ordination with environment" (Tharumarajah *et al.*, 1996). Each holon combines its set of competencies with its lateral partners, with whom co-operate in order to achieve both individual and system goals. This suggests the holon is an autonomous entity, including operational features and individual goals. System goals and plans are partially defined in higher holons and travelling down the holarchy, they are progressively refined. The result is a highly complex but reactive system with minimal concession to efficiency.

### 2.4. FF Vs. BMS Vs. HMS

Fractal Factory, Bionic Manufacturing Systems and Holonic Manufacturing systems are three examples of organisational paradigms comprehended in the broad term of Open Hierarchical Systems introduced by Koestler in 1967. These paradigms suggest the idea that manufacturing systems, despite the increased autonomy required by individual entities, it will continue to need a hierarchical structure in order to maintain the overall system coherence and objectivity, and guarantee the inter-entities conflict resolution arising from the individual and autonomous attitude of the entities. The three paradigms have different origins, which imply different technical approaches. However, some points are common to the three: (i) composed by smaller, autonomous, and competitive entities; (ii) co-operating and grouping entities; (iii) hierarchical structures; (iv) dynamic nature of structures; (v) goal oriented grouping. These paradigms were proposed in order to provide the enterprise with standard and effective organisation and technological approaches, which does not mean that hybrid approach could not be developed or combined, forming heterogeneous structures, exploiting maximum benefit in each particular application.

## 3. Functional Services

From the conceptual models described above, through its complete realisation, a long way must be tracked in different research subjects, especially in organisational infrastructures and knowledge management. 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, Security and Directory conveniences. The following service intends to supply such conveniences.

### 3.1. The Information Service

The Information Service (InfSys) combines the Entity, Security and Directory conveniences. To describe it, its conveniences will be separated in the Identification and the Directory sub-services.

The Identification service concerns the inclusion into, and exclusion off the system. 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. At the moment, it comprehend a bilateral authentication based on public/private key;
3. Registration. This phase correspond to the entity admission to the system. It corresponds to insert its 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 contact any other member, and vice-versa.

In the Unregistration phase the entity is excluded from the system. 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. Another reason to unregistration is concerned to Directory service and will be referred below. When the unregistration occurs the remaining members are informed about the fact and requested to discard any active conversation with the unregistered entity.

Concerning the Directory services, the InfSys intends to provide the community with domain public information, namely capabilities one wants advertise in order to be requested, e.g. performing operations, reliability, 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 are as follows:

- Information has a single owner/publisher, who has to be is a registered member;

- The owner is responsible for information management (publish, update and unpublished);
- Only InfSys provides published information to the community (if any entity provides such information individually, the entity is unregistered);
- Owner limits information time validity. Information can not be updated/unpublished until the supplied information loses validity;
- 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 unique connection to system, information coherence and validity are easier to maintain since no duplicate or ambiguous identification of information occurs. Members are free to use this service even if it is strongly recommended.

### 3.2. The Pooling Service

The Pooling Service (PoolSys) concerns with message delivery to entities temporarily unreachable or unregistered in InfSys. The PoolSys 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 as any other capability. In order to achieve reliability (even it is not frequently demanded), multiple PoolSys entities are allowed in the system even no co-ordination or co-operation is required. Any message is stored in PoolSys entity DB without any processing. PoolSys entity should periodically query its associated InfSys for the destination entities and when available, send it the message.

### 3.3. The Domain Service

Described organisational paradigms propose dynamic grouping during operation. 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, considering the above mentioned fractal factory project oriented structure; each factory entity represents a potential project 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 (DomainSys) provides conveniences for domain management. To type of entities are recognized, the creator/manager, and the entities who populate the domain. The developed 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 domain InfSys 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 primitive entity to generate passwords to the entity and deliver it to the entities prior to the domain creation. Hence, it is possible to the new entity contact and securely authenticates entities. Another solution is the synchronisation of addresses and identification between creator and its child. 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. However, excepting the well-known permissions mechanism (similar to that of O.S.) no other concurrent mechanisms

were studied. 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.

### 3.4. Application example

Additionally, it worth mention the fact the conversations messages uses the KQML specification, though there is no need to manipulate these messages, since the supplied API entirely encapsulates protocol complexity. The API is developed in Erlang, a functional 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 both provide and require these services. Specific application behaviour is developed as additional layer. Figure 1 represents this approach.

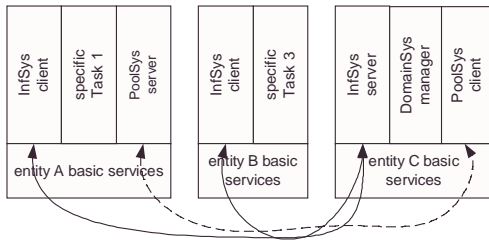


Figure 1 - Functional Services Implementation

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 was to call 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 between others domain requested name, InfSys server name, broadcast port number and various timeout values.

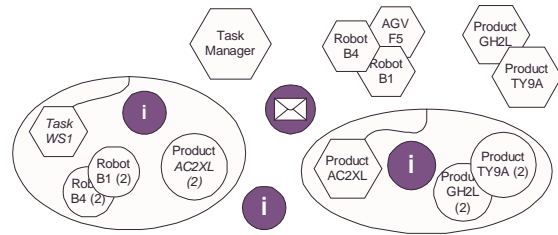


Figure 2 – Example of domain representation

The hexagonal forms represent primitive entities and circle forms represents derivate entities. Notice that Product AC2XL is primitive entity composed by two derivate entities. At some point in time, a new order arrived 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 prior example represents the arguments to create a new domain. The *domain\_manager* argument defines the manager execution module, function and its initial arguments. The *domain\_entities\_allowed* defines the entities which authentication based in password is not necessary. Finally, the *domain\_entities\_mandatory* argument specifies the entities that participation is mandatory.

## 4. Knowledge Sharing project

In the section it will be described 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. KQML/KIF, ACL and other agent-based communication languages provide primitives to assist conversation set-up, information and knowledge exchange in a statically predefined form. However, the dynamic nature of the virtual organisations requires a more automatic configuration process that allows the exchange and common

understanding 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. 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 related problem; only postpone it. During the last years, came up a large number of ontologies about different domains, which emphasises the incompatibilities problems between ontologies. A proposed solution is the translation between different ontologies, performed by specific translators. As these translators become a place of translation between several ontologies, it turns into a *lingua franca* (Uschold and Gruninger, 1996). However, in most cases these translators are hand written, implying 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 to automatically define and negotiate ontology translation procedures. This translator applies different AI techniques like Reasoning and Learning, with other less AI dependents, like Negotiation strategies, Natural Language processing and Dictionary and Thesaurus features, etc. Additionally, dealing with all the numerous representational approaches might cause the most laborious problems.

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 3 represents the proposed approach.

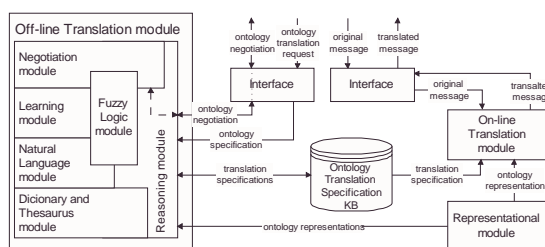


Figure 3 – Proposed approach schema

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 were 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, and negotiations (Negotiation module) and compromises (Fuzzy Logic module) between parts are required;
- 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.

The most common ontology architectures are composed by multiples layers, each inheriting and specializing multiple lower ones (Uschold *et al.*, 1998), creating even more detailed one (Figure 4).

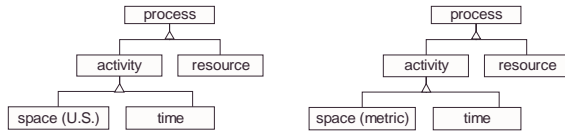


Figure 4 – Inheriting and specialising ontologies.

Additionally, 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. 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. The holonic, bionic and fractal systems, views as technological-organisational approaches were briefly described and compared which allowed finding common characteristics and behaviour patterns and in some sense, advocate these paradigms are not always contradictory and can even be combined into specific systems. Considering the identified patterns, it was considered beneficial or even necessary to provide system specific services concerning Entities, Directory, Grouping, Communication and Security management, provided trough tree complementary services: The InfSys, PoolSys and DomainsSys services. Its implementation and utilisation topics were presented.

In addition, a brief analysis about inter-domain knowledge sharing demonstrated the need to improve functional services, in this particular to the automatic construction of shared and accepted ontologies. Considering the analysed paradigms, namely their dynamic nature, agility and heterogeneity, the use and prior agreement about shared ontology is considered a very promising improvement vector. However, the proposed approach needs important experts' feedback and fundamentally intensive experiments. Apparently, a reduced amount of research is running on this topic.

## 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>.

Ontology.Org, 1999; The need for a shared ontology; Ontology.Org; 1999; <http://www.ontology.org/main>.

Parunak, 1998; What can Agents do in Industry, and Why? An Overview of Industrially-Oriented R&D at CEC, H. Van Dyke Parunak, Industrial Technology Institute, Forthcoming at CIA98.

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>.

Sihn, 1997; The Fractal Factory: A Practical Approach to Agility in Manufacturing; Wilfried Sihm; Proceedings of The Second World Congress On Intelligent Manufacturing Processes & Systems, pp. 617-621, Budapest, Hungary, June 10-13, 1997.

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, n° 3, 217-226.

Uschold and Gruninger, 1996; Ontologies: Principles, Methods and Applications; Mike Uschold and Michael Gruninger; Knowledge Engineering Review; Volume 11 Number 2, June 1996.

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.