

SMART CONFIGURATION OF DYNAMIC VIRTUAL ENTERPRISES

R. J. Rabelo, F. Baldo, R. Tramontin Jr., A. Pereira-Klen, E. R. Klen
*Federal University of Santa Catarina, Department of Automation and Systems, GSIGMA -
Intelligent Manufacturing Systems Group
Florianópolis (SC), BRAZIL
rabelo@das.ufsc.br
{baldo, tramontin, klen, erklen}@gsigma.ufsc.br*

This paper presents an approach to configure dynamic virtual enterprises supported by smart tools that are implemented to facilitate and to speed up this process in a low cost and effective way. The configuration comprises the setup of the information to exchange among the VE partners during the VE life cycle, the integration of their legacy systems with the VE platform, the configuration of the VE topology, and the definition of information access rights for the VE partners. Details about the implementation of the supporting tools as well as some conclusions based on the results achieved are also given.

1. INTRODUCTION

A Collaborative Networked Organization (CNO) is a dynamic, temporary and logical aggregation of autonomous entities (companies, people, governmental institutions, etc.) that interact with each other as a strategic answer to attend a given opportunity or to cope with a specific need, and whose operation is achieved by the coordinated sharing of skills, resources and information, totally enabled by computer networks. This work focuses on Virtual Enterprises (VEs), which is seen as an instance of CNOs. The partners of a VE may be involved in multiple VEs simultaneously – being or not interdependent VEs – and can join or leave the VE along its life time.

The VE accomplishment is carried out along some phases representing its life cycle, namely Creation, Operation & Evolution, and Dissolution (Spinosa et al., 1998) (Camarinha et al., 1999). Some work has been developed for dealing with these phases at different levels of depth and scope. However, a number of issues remain almost untouched as the VE area has greatly evolved in the last years and the application of its concepts became more mature and understandable.

Current needs of the VE area demand agility (Rabelo et al., 1999). In the Creation phase this means that the VE should be ready to operate as fast as possible. However, the configuration of all the required information in the VE platforms is

usually done manually, without system assistance, implying in a slow installation and configuration processes, subject to many errors and hence costly.

In this sense, this paper explores some issues here generally called as *configuration* aspects. They involve activities since the setup of VE supporting platforms until their final instantiation by “real” VEs, after which the VE is ready to operate. More specifically, these issues comprise i) the setup of the information to be exchanged among the VE partners during the VE life cycle, ii) the integration of VE partners’ legacy systems with the VE platform, iii) the configuration of the VE topology, and iv) the definition of the information access rights for the VE partners. Three tools have been developed to support these configuration issues and they will be introduced later on.

The configuration of dynamic VEs considers that the members’ roles, the information access rights, the business processes specification and even the VE members are neither predefined nor fixed. Besides that, it is relevant to consider that all these information can change along the time and they should be efficiently updated and put available to the VE participants, especially to the VE coordinator, so that the partners can work and react based on reliable and timely information.

The proposed work presents the configuration activities as integrated system functionalities implemented in a VE management system - called *VE Cockpit* - with different levels of automation, towards an automatic configuration of VEs. The VE Cockpit is an interactive and distributed system that helps the VE coordinator in the management of the VE’s business processes based on the principles of on-line business intelligence, covering functionalities that offer some support for the VE setup, creation, operation and evolution (Rabelo et al., 2002) (Rabelo et al., 2004).

This paper is organized as follows: Chapter 2 briefly presents the global VE configuration framework subject of this research, positioning chapters 3 and 4. Chapter 3 details the work developed in the preparation of the VE setup, and Chapter 4 details the configuration of the VE topology and information visibility. Chapter 5 shows some implementation results of the proposed integrated configuration framework. Chapter 6 describes the main conclusions.

2. VE CONFIGURATION FRAMEWORK

The two macro configuration activities involved in this work – VE Environment Setup and VE Configuration – are comprised in an integrated framework (figure 1). These activities are carried out in four steps (in gray in the picture) and they are supported by three software tools. The VE creation phase is extended and gives rise to a prior stage called “VE Environment Setup”. Both activities are seen as a part of what nowadays is being called “Breeding Environment” (Camarinha et al., 2004).

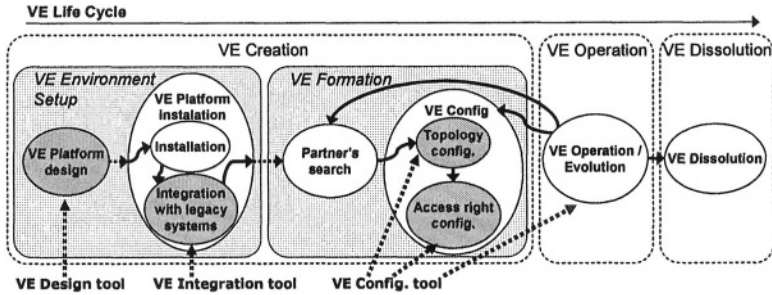


Figure 1 – VE life cycle framework in the VE Cockpit system

The VE Environment Setup corresponds to a “preparatory” activity that is concerned to the design of the VE platform¹ itself and its final installation at the enterprises. In the first step the specification of the “reference information models” and final setup of the VE processes are defined and implemented. The reference models indicate the information to exchange with the participants. In the second step the VE platform is installed at the enterprises and integrated with their legacy systems (Tramontin, 2004).

The VE Cockpit system, as one of the VE platform’s modules, uses a “reference interface” associated with the mentioned reference models. It means that the VE Cockpit system, and hence the configuration functionalities, are designed/open to be integrated to “any interoperable” VE platform. Therefore, the VE Environment Setup activity is to be carried out only once no matter how many VEs a given enterprise will participate in.

The next stage of configuration is related to the VE creation, when a VE will be really established. In this work the activity of partners’ search and selection (PSS) will not be addressed although a supporting tool for that has also been developed inside the scope of the projects previously mentioned (Schmidt, 2003) (Rabelo et al., 2004b). It is assumed here that a given set of enterprises was found and selected to make part of a VE and issues like trust building and legal aspects are overcome. However, regarding reference information models, the most important here is to consider that the elected PSS tool should be open to receive information from the VE platform setup activity as well as to provide the required information to configure the VE topology and the access rights.

The roles (e.g. supplier, retailer, main producer) of every partner (i) and the links between them (ii) are specified in the VE topology configuration according to the material/information flows of the VE, right after the information access rights of the partners are defined. Actually, these two steps can also be executed during the VE Evolution phase, where changes in the VE may occur, requiring modifications in the topology and/or in the level of information visibility (Baldo, 2003).

3. VE PLATFORM SETUP

3.1 Configuration of the Reference Information Models

A key aspect in this work to support the envisaged automatism is the assumption / notion of reference models (Tolle et al., 2002), an essential issue to model the information to exchange among the VE members, regarding their current roles in the value chain. This information refers to the enterprises' data (identification, address, etc.) as well as to the data associated to the VE's main processes, namely *sales*, *manufacturing* and *delivery* (according to the SCOR model [www.supply-chain.org]).

One problem behind this modeling is the need to agree on the syntax, semantics and the structure of the information to exchange, which here is made in the VE platform design (see figure 1). The achievement of this agreement usually takes a very long time, with progressive refinements and validations in the models until their final definition. It means that when the VE platform is tested every change in any element of the involved information structures requires a redefinition/recoding of parsers as well as of the database's tables, relations and supporting access functions, implying in a very hard, time-consuming and consequently costly process. The approach here applied aims at facing that problem, providing a semi-automatic way to assist in this design phase.

A reference information model defines a "standard" model for each of those VE processes for specific application domains. Despite the number of initiatives being carried out nowadays in that direction (e.g. Rosettanet [www.rosettanet.org], ebXML [www.ebxml.org], CIMOSA [cimosa.cnt.pl]), there is a lack of a common reference model for VEs. This work does not intend to provide this missing model, but rather to offer an effective way to configure VEs (in the VE Cockpit system). In this work a "homemade reference" information model called Distributed Business Process (DBP) model (Pereira-Klen et al., 2001) was used and it has been complemented along a number of projects (Prodnet-II (www.uninova.pt/~prodnet) [97-99], Damascos (www.inescporto.pt/~damascos) [00-01] and MyFashion (www.myfashion.org) [02-04]).

The key technology applied to support this approach is the XML language (www.w3.org/XML). XML has emerged as the standard representation language for information exchange (especially) among heterogeneous systems, providing a concrete and formal basis for systems interoperation. Therefore, all the information structures of the reference model are expressed in XML, modeled in the form of *DTDs* (Document Type Definition) or *XML Schemas*, easily validated by XML parsers. The XML data binding technique used for code generation from XML schemas (Bourret et al., 2003) speeds up the configuration process in the design step. The developed tool (*VE design tool*) takes each XML schema (corresponding to the reference information model) and generates *automatically* the related classes and the database tables. These classes implement methods to load/save their contents from/to XML documents, using a XML parser to validate them. Further, they implement the database access methods in SQL (figure 2). After this, the VE application can be tested. This process repeats until the final reference models to be used by the VE platform / its applications are settled. After each test / modification, the user basically needs to introduce the new reference model (in XML) in the VE

design tool and it generates the necessary information structures (classes/tables) without any recoding.

This tool can also be used by the other modules of the VE platform, other than the VE Cockpit system.

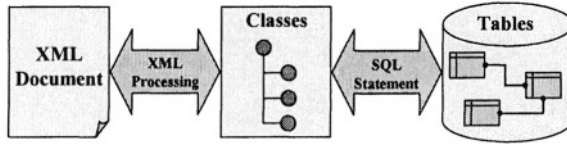


Figure 2 – Conversion process of XML, Classes and database Tables

3.2 Integration with Legacy Information

The VE partners should exchange between them information about the business they are involved in order to fulfill VE objectives. This information usually comes from their databases. Therefore, it is mandatory to integrate these local information repositories so that the data can be accessed and further sent out. The ideal situation is the case when all the partners use the same reference information models and their ERP systems / databases are equivalent in terms of information they can provide. However, this is not a realistic assumption regarding their natural heterogeneity. Given this scenario, XML arises again as a suitable technology to allow the partners' systems interoperability.

It is to be highlighted that the focus here is the integration of the information and not the integration of the processes, which is often provided by workflow systems. Another important aspect regarding dynamic VEs is the assumption that there is not a centralized VE database nor time to configure powerful federated schemas in distributed databases (as the ones used in Frenkel et al., 2000). The approach applied in this work is less powerful but it is efficient in spite of the data redundancy naturally present. The goal is to integrate the different partners' databases, normally used as the information repositories that stores most of the information generated by their legacy systems. This integration activity uses to be manual and is also costly, complex and long, but vital for the correct functioning of the VE platform. The developed tool (*VE Integration tool*) implements this step in a semi-automatic way, with some computer assistance, applying the *XML middleware* concept (Bourret et al., 2000) (Sousa et al., 2003) (Shanmugasundaram et al., 1999).

The integration activity is done when the VE platform is installed in each partner. Once the supporting database is identified an interactive process driven by the *VE Integration tool* is started to establish a link between every field of the reference information model with the legacy information models. Afterwards a relational and semantic *mapping* is created generating automatically and dynamically the SQL code for the information access (figure 3). In practical terms this process allows a rapid preparation of a partner to receive and to send the needed information when requested by some of the VE platform's modules and/or partner.

After the execution of these two configuration steps (3.1 and 3.2) an enterprise would be ready to participate in a VE.

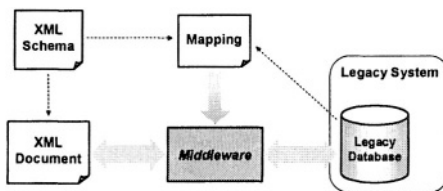


Figure 3 – Conceptual model of information integration

4. VE CONFIGURATION MODEL

While the two configuration steps addressed in the previous chapter are seen as preparatory activities for the VE operation (see box “VE Environment Setup” in figure 1), this chapter stresses two other configuration steps that are executed once the VE partners are identified (see box “VE Formation” in figure 1). These two steps refer to the configuration of the VE topology and of the VE access rights, and they are carried out by the *VE Coordinator*. The VE Coordinator is who manages the entire VE, coordinating the execution of the VE’s distributed business process close to the *VE members*.

4.1 VE Topology Configuration

Once the VE partners are defined the VE Cockpit system should be configured according to the material and information flows among the partners. The configuration of the VE topology means to specify: i) the partners names ²; ii) the partners roles; iii) the links between the partners, and ii) the business process under responsibility of each partner. In general, this can be seen as the *VE plan*. Likewise the other macro activity (stressed in the chapter 3), this configuration is usually done manually, without system assistance, taking too much time, especially if dynamic VEs are considered.

In this work an automatic approach is applied, implemented by a tool called *VE Configuration*. The VE Cockpit system offers a way where each of its “peers” installed in the partners is enabled to get the required information from their legacy databases (*VE Integration tool*) and to send them to the VE coordinator. The VE topology is automatically generated at the end of this process (figure 6) and it is further exhibited to the partners according to their level of information visibility (see next section).

4.2 Access Rights Configuration

The configuration of the access rights is an essential issue in VEs. On the one hand, regarding information security, privacy, confidentiality, among others aspects, the partners only feel confident to participate in the VE if there is some protection in their private data. On the other hand, the realization of the VE concept involves the sharing of information. The main approach being applied on this problem is to provide means to configure levels of information visibility among the VE’s partners (Pereira-Klen et al., 2001). This work provides an interactive method and tool (the *VE Configuration tool*, the same one used in the previous step) to support that

configuration. More than this, this tool allows dynamic reconfiguration as the VE state can change along the time.

The configuration is carried out at two levels. The first level defines who can “see” who in the VE, i.e. the partners’ visibility. This can vary from VE to VE, but the VE coordinator is the only one who can always see the whole VE partners. The minimum degree of visibility allows only a first-tier vision. In the other extreme, all the VE partners can see all the other involved partners. This configuration is executed by the VE coordinator according to a previous agreement established with the VE partners, assuming a trust-based environment.

In the second level the VE coordinator specifies which information (from the reference information model) each VE partner will be allowed to see (see figure 7). In a centralized way, the VE coordinator receives (via the *VE Configuration tool*) all the required information, filters them according to partners’ *visibility agreement*, and sends this information to the partners accordingly. This is done based on the concept of “supervision clauses”, an auxiliary information structure aggregated to the VE contract (Rabelo et al., 1999) (Pereira-Klen et al., 2001). By means of an interactive process the VE coordinator checks or unchecks every reference information model’s field (“box”) in such a way (s)he knows exactly which information can be delivered to each partner. It also takes into account the real availability of the partners’ information that was mapped during the integration with their legacy systems (see section 3.2). The VE coordinator can modify that configuration on the fly, without any recoding, according to new requirements of visibility criteria.

5. IMPLEMENTATION AND RESULTS

The three developed tools are agent-based systems which are part of the VE Cockpit system. Both the multi-agent infrastructure (MASSIVE platform: www.gsigma.ufsc.br/massyve) and the *VE Configuration tool* are implemented in C++. The *VE Design* tool and *VE Integration* tool are implemented in Java.

In terms of global architecture, there is one VE Cockpit system’s peer installed in each VE partner (figure 4). The system is composed of a number of agents (associated to the system’s functionalities) and a (relational) database. From the partner’s side, there is the “legacy” database with which the system will interact. The system recognizes automatically when a given partner is the VE Coordinator, and therefore only the peer installed at the VE coordinator is fully enabled with all the system’s functionalities.

The interaction among the agents and the system’s peers is supported by a communication infrastructure (called *Message Management System - MMSys*) based on an open/“standard” XML/CORBA interface. If a VE platform is driven by a workflow system, for instance, then it should interact with the MMSys. The system’s functionalities are all available via user-friendly graphical interfaces.

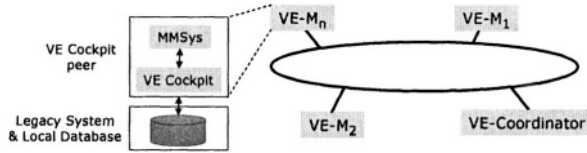


Figure 4 – Global Architecture

Regarding the three tools that implement the four configuration steps (depicted in the chapter 3 and 4), they are independent but can work in an integrated way as in the VE Cockpit system. Figure 5 shows an interface of the *VE Design tool*, which assists the configuration of the information to exchange with VE partners.

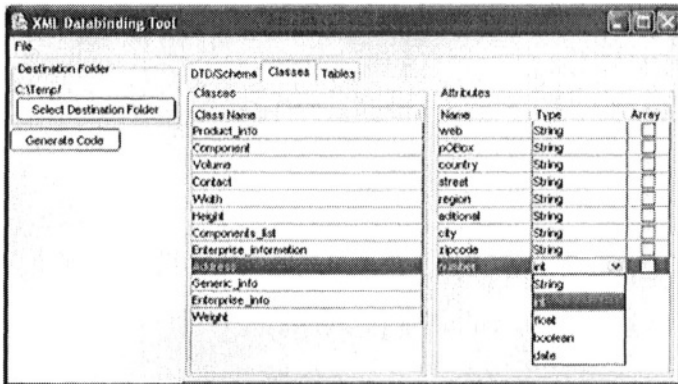


Figure 5 – Configuration of the information to be exchanged

Figure 6 shows an interface of *VE Integration tool*, which enables the integration with the partners' legacy systems. It presents the reference information model previously agreed and the semantic and relational setting against the local database.

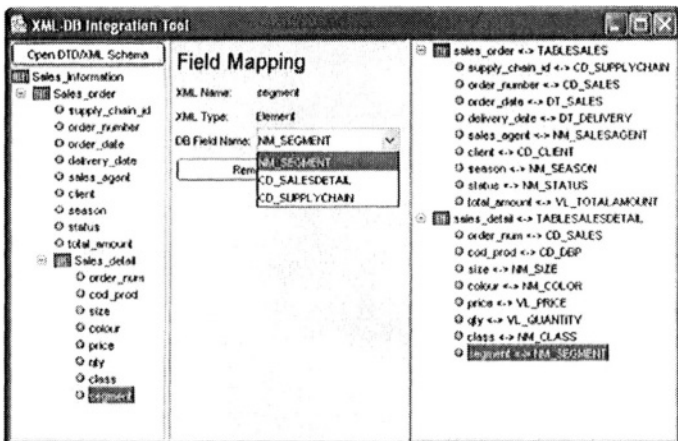


Figure 6 – Integration with the legacy systems

Figure 7 illustrates the main graphical interface of the topology configuration (*VE Configuration tool*), where each partner has a particular visibility level. For instance, the VE coordinator can see all the VE partners whereas Member 2 only sees some of them.

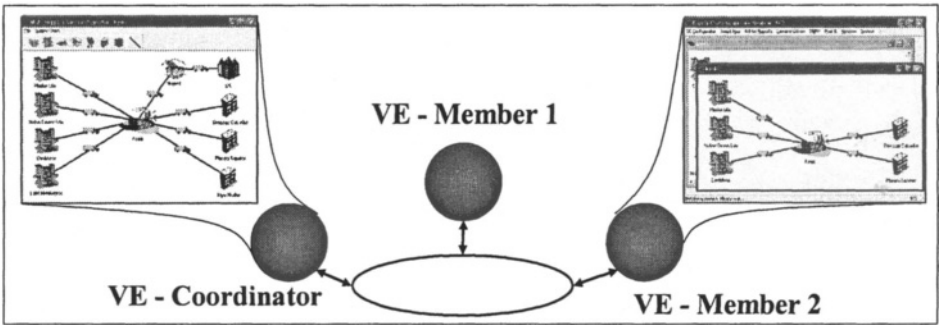


Figure 7 – VE topology & Information visibility configuration scenario

Figure 8 shows the main graphical interface of *VE Configuration tool* used by the VE coordinator to configure information access rights of every partner.

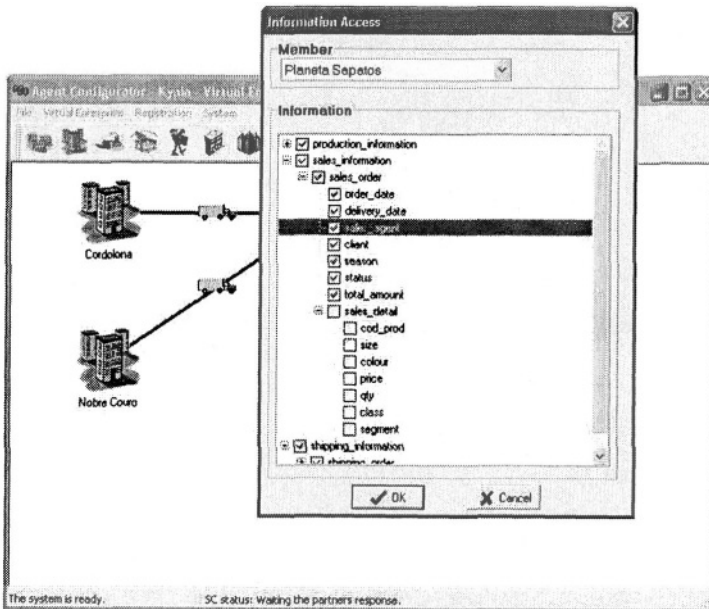


Figure 8 – Configuration of the information access rights

6. CONCLUSIONS

This paper presented some results of the investigation being made on how dynamic VEs can be rapidly configured in order to be ready to operate as fast as possible. As the VE configuration is usually carried out manually, without computer assistance, it

is subject to many errors that can jeopardize the adequate VE functioning. Additionally, a manual process uses to take much time and to be very costly.

As a contribution to solve this problem, an approach that automates some essential steps of the VE configuration phase was proposed. These steps involves: i) the specification of the information to exchange among the VE partners during the VE life cycle; ii) the integration of VE partners' legacy systems with the VE platform; iii) the identification of the VE topology; and (iv) the definition of information access rights for the VE partners.

Three integrated and interactive tools were implemented to support these steps, with different levels of automation. They are modules of a wider VE management system called VE Cockpit. Multi-agent systems, CORBA, and mainly XML, are the key information technologies utilized by these tools.

Actually, the configuration scenario covered in this work comprised not only the preparatory aspects to support the VE setup and VE formation, but also the VE operation/evolution phases when a reconfiguration is needed in the VE Evolution phase.

This work is strongly based on the notion of reference information models. This issue needs much more investigation in spite of the number of ongoing international initiatives working on it. As there is no common reference model for VEs that can be used in several domains, this investigation worked upon a homemade reference model conceived along some international projects. This model served to test and to evaluate the potentiality of the approach to speed up and to increase the quality of the configuration process, providing a more concrete basis to guide next developments.

Regardless a number of aspects such development environment, the experience of the development team, the time every partner takes to evaluate internally the required information to exchange, among others, the time average estimated that are saved in the VE Environment Setup (figure 1) activity is up to 50% compared to a non-assisted process. The VE Configuration activity (figure 1) is more difficulty to estimate a quantitative number. Its main benefit in felt during the VE creation and evolution phases, obtained thanks to the interactivity/semi-automatic execution, offering a more trustful system in terms of guaranteeing the correct information visibility of the VE partners.

Acknowledgements

This work was partially funded by CNPq (the Brazilian Council for Scientific and Technologic Development). This work is being developed under the scope of the following projects: IST MyFashion.eu (www.myfashion.org), IFM (www.ifm.org.br) and Capes-Cofecub (www.das.ufsc.br/gia/capes-cofecub).

7. REFERENCES

1. Baldo, Fabiano, An Approach for a Semi-Automatic Configuration of Virtual Enterprises – A Multi-agent Approach [in Portuguese], Master Thesis Dissertation, Federal University of Santa Catarina, Brazil, 2003.
2. Bourret, R.; Bornhövd, C; Buchmann, A., A Generic Load/Extract Utility for Data Transfer between XML Documents and Relational Databases. In P. You, ed., Proc. **2nd**

- Intl. Workshop on Advanced Issues of E-Commerce and Web-Based Information Systems, pp. 135-143, 2000.
3. Bourret, R.; 2003. XML Data Binding Resources, <http://www.rpbouret.com/xml/XMLDataBinding.htm>, accessed in April 15th, 2003.
 4. Camarinha-Matos, L. M.; Afsarmanesh, H., The Virtual Enterprise Concept. Proc. PRO-VE'99, pp. 3-14., Kluwer Academics Publisher, 1999.
 5. Camarinha-Matos, L. M.; Afsarmanesh, H., Supporting Infrastructures or New Collaborative Forms, in Collaborative Networked Organizations, pp. 175-192, Kluwer Acad. Pub., 2004.
 6. Frenkel, A.; Afsarmanesh, H.; Garita, C.; Hertzberger, L., Supporting Information Access Rights and Visibility Levels in Virtual Enterprises, Proc. PRO-VE'2000, pp. 177-192, Brazil, 2000.
 7. Pereira-Klen, A. P.; Rabelo, R. J.; Ferreira, A. C.; Spinosa, L.M., Managing Distributed Business Processes in the Virtual Enterprise, in Journal of Intelligent Manufacturing, V. 12, N. 2, pp. 185-197, 2001.
 8. Rabelo, R. J.; Camarinha-Matos, L. M.; Afsarmanesh, H., Multiagent-based Agile Scheduling, in Robotics and Autonomous Systems, North-Holland, N. 27, pp. 15-28, 1999.
 9. Rabelo, R. J.; Pereira-Klen, A., Business Intelligence Support for Supply Chain Management, in Balanced Automation Systems in Manufacturing and Services, Kluwer Academics Publishers, pp. 437-444. 2002.
 10. Rabelo, R. J.; Pereira-Klen, A.; Klen, E. R., Effective Management of Dynamic Supply Chains, to appear in International Journal of Networking and Virtual Organisations, 2004a.
 11. Rabelo, R. J.; Pereira-Klen, A.; Spinosa, L. M.; Ferreira, A. C., Agile Supply-Chain Coordination in the Virtual Enterprise Environment, Proc. IV Brazilian Automation Symposium, pp. 289-294, 1999.
 12. Rabelo, R. J.; Wangham, M. S.; Schmidt, R.; Fraga, J. S., Trust building in the creation of virtual enterprises in mobile agent based architectures, in Processes and Foundations for Virtual Organizations, Kluwer Academics Publishers, pp. 65-72, 2004b.
 13. Schmidt, R., Partners' Search and Selection in Virtual Enterprises – A Mobile Agent Approach [in Portuguese], Master Thesis Dissertation, Federal University of Santa Catarina, Brazil, 2003.
 14. Shanmugasundaran, J.; Tufte, K.; He, G., Relational Databases for Querying XML Documents: Limitations and Opportunities, Proc. 25th International Conference on Very Large Databases, Edinburgh, Scotland, 1999.
 15. Sousa, L.; Souza, F., A Generic Middleware to Migrate Data between Relational Database and XML Documents, Proc. 3rd IFIP Conference on e-Commerce, e-Business, and e-Government, São Paulo, Brazil, 2003.
 16. Spinosa, L. M.; Rabelo, R. J.; Pereira-Klen, A., High-Level Coordination of Business Processes in a Virtual Enterprise, in Globalization of Manufacturing in the Digital Communications Era of the 21st Century, Kluwer Academics Publishers, pp. 725-736, 1998.
 17. Tolle, M.; Bernus, P.; Vesterager, J., Reference Models for Virtual Enterprises, in Proc. PRO-VE'2002, pp. 3-10, Portugal, 2002.
 18. Tramontin Jr., R. J., Data Configuration and Integration in Platforms for Virtual Enterprise [in Portuguese], Master Thesis Dissertation, Federal University of Santa Catarina, Brazil, 2004.

¹ VE platform is the set of integrated software modules designed to comply with the business processes that are required to support all transactions among the VE partners (e.g. collaborative design,

EDI, information management), including a communication infrastructure for the interactions inside the platform, among the partners and with their legacy systems.

² If the VE coordinator has a partner's search and selection (PSS) tool and if it follows the expected reference information model then the developed tool receives the partners' names from PSS.