

# 4 COORDINATION AND CONFIGURATION REQUIREMENTS IN A VIRTUAL ENTERPRISE

L.M. Camarinha-Matos <sup>♦</sup>, C. P. Lima  
New University of Lisbon, Portugal

*Coordination is a determinant element in Virtual Enterprises. Proper coordination policies supported by flexible coordination mechanisms are necessary to ensure the cooperation among partner enterprises. The large diversity of scenarios and operating policies suggest a flexible and configurable approach to coordination. The supporting mechanisms and approaches developed in the workflow management and business process modeling areas are among the most used in this domain. Finally the motivation for a hierarchical coordination approach is introduced.*

## INTRODUCTION

### Concept of coordination

Cooperation is a key underlying aspect in the Virtual Enterprise (VE) paradigm. A VE is a temporary consortium of autonomous enterprises that decide to cooperate in order to achieve a common goal (i.e. to jointly respond to a business opportunity). This cooperation involves, in general, sharing information and other resources, communication, “producer/consumer” relationships (information or product-related), and collaborative activities.

Activities carried out by a company are usually organized in “clusters” of inter-related activities called *processes* (*business processes*). Each process is designed in order to achieve a (partial) specific objective. When properly “orchestrated”, the combination of various processes will lead to the achievement of a global VE goal. An obvious difficulty here comes from the fact that independent actors (different members of the VE) that have different behaviors, different (and even competing)

---

<sup>♦</sup> Corresponding author address: Universidade Nova de Lisboa, Quinta da Torre, 2825 Monte Caparica, Portugal, tel. +351-1-2948517, fax +351-1-2941253, e-mail: cam@uninova.pt.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: [10.1007/978-0-387-35577-1\\_37](https://doi.org/10.1007/978-0-387-35577-1_37)

priorities and motivations, and different perceptions of the environment might perform these processes. Therefore, an effective cooperation requires coordination. Coordination is necessary for preventing “anarchy” or “inefficiency” (chaos) in what respects achieving a common global goal.

Coordination represents a new and multidisciplinary research area that finds its roots on contributions in many other disciplines such as computer science, organization theory, economics, linguistics, psychology, artificial intelligence, etc. In fact, in recent years, there has been a growing interest for coordination theory and coordination mechanisms (Malone and Crowston, 1997), (Papadopoulos and Arbab, 1998), (Schmidt and Simone, 1996), (Simone et al., 1996), namely when discussing the activities performed by complex and distributed systems. Different authors have proposed a number of definitions for coordination. A sample of these definitions:

“*Coordination is managing dependencies between activities*” (Malone and Crowston 1994).

“*Coordination is the process of building programs by gluing together active pieces*” (Carriero and Gelernter, 1992).

“*Coordination can be defined as the study of the dynamic topologies of interactions among interaction machines, and the construction of protocols to realize such topologies that ensure well-behavedness*” (Arbab, 1998).

“*Coordination is a property of interaction among some set of agents performing collective activities*” (Bond & Gasser, 1988).

According to (Papadopoulos and Arbab, 1998) and (Krone *et al*, 1998), a coordination model is a triple (E, L, M) where E represents the entities that are the subject of coordination, L is the media used to coordinate the entities, i.e. the actual space where the coordination takes place, and M is the semantic framework or the coordination laws the model adheres to. Applying this framework to an infrastructure for VEs, E represents the VE members or the tools / modules / services available in each enterprise. L represents the communication infrastructure adopted in this distributed environment, including the set of valid messages exchanged among VE members. M represents the rules that dictate the behavior of each member towards the VE and the behavior of the VE as a whole according to the cooperation contract established by the consortium.

The coordination needs do not arise only among the activities / processes being performed by different VE members; there is also a need for coordination among the activities / processes taking place inside each VE member enterprise, as enterprises are “represented by” many applications / tools.

It is also important to notice that *heterogeneity* is a key characteristic in VE coordination scenarios. The computational infrastructures found in each VE member, the used tools and the corporate culture and business practices are quite diversified. Any practical coordination mechanism has to support the *interoperation* of such diverse environments.

### **Flexibility and configuration**

One interesting aspect in VE coordination is that VE is a new paradigm for which there is not yet a well-established set of representative experiences. As a consequence, there is not much knowledge about the way VE systems behave or how

they could evolve. In fact it is expectable that the new possibilities offered by technology will induce new ways of “working”, i.e. new business practices are likely to emerge requiring new forms of coordination.

Various research initiatives have been attempting to model the VE “behavior” but this work can be considered in its infancy. For instance, the Esprit VIVE project proposes a detailed list of business processes for the creation and operation phases of a VE (VIVE, 1999). The Esprit PRODNET II also analyzed various scenarios in order to identify the main business processes and their coordination needs. Similar analysis works are being performed by many other initiatives. It is however clear that we face a large diversity of scenarios and VE organizations, each one requiring specific behaviors. Even inside a VE different behaviors regarding the same type of event may be observed in different enterprises. And these behaviors are likely to evolve as a consequence of the acquired experiences with cooperation and as a result of the trust building processes. Therefore, an important requirement for a coordination approach in VE is *flexibility*.

One interesting approach, typically found in the workflow management systems (Lawrence, 1997), and more recently being also proposed by other works on coordination languages (Papadopoulos and Arbab, 1998), is the separation between *coordination* and *computing* or *service processing*. The computing part comprises the application-oriented functionality and the coordination part covers the communication and management of dependencies among activities (sequencing, synchronization, conditioned transitions, etc.). This general schema contributes to achieve the desired flexibility: the data processing functions (the more stable part) are organized in libraries of services whose invocation is managed by a separate coordination plan (Camarinha-Matos *et al*, 1997b), (Camarinha-Matos and Afsarmanesh, 1997). From the coordination point of view, services are seen as black boxes with well-defined interfaces. Changes in behavior are implemented by just changing the coordination part.

The need for *configuration* is a closely related requirement for a flexible coordination system. During the set up phase it is necessary to configure the VE environment, and the infrastructure located at each VE node in particular, to reflect both the general contractual agreements among VE partners and the levels of autonomy and privacy required by each enterprise. During the VE operation additional (re-)configuration steps may be required to cope with the evolution in the VE and / or the evolution of its behavior. Coordination flexibility and configurability are, therefore, key requirements for a VE supporting infrastructure.

## COORDINATION MECHANISMS

### Contributions from various areas

The development of coordination mechanisms in computer science started long before this area has shown signs of emerging as a quasi-autonomous discipline. Early coordination mechanisms can be traced back to the initial works on operating systems and real time concurrent programming. Another important contribution can

be found in the area of Petri nets and its many dialects, specially when this formalism started to be applied not only as a modeling and analysis tool but also as a control mechanism. More recent contributions to coordination mechanisms can be found in the areas of workflow management systems, computer supported cooperative work (CSCW), business process modeling, multi-agent systems (MAS), and coordination languages.

*Petri nets* represent a tool for modeling systems with interacting and concurrent components. It represents an analytic and graphical approach useful for modeling, analyzing and simulating discrete event dynamic systems. A Petri net  $\{P, T, I, O, M\}$  is a directed graph consisting of two types of nodes, places (P) and transitions (T), where arcs are either from a place to a transition (I) or from a transition to a place (O). In graphical representation, places are drawn as circles and transitions as bars. A marking (state) assigns to a place a non-negative integer (number of tokens).

A place may represent a process or activity (or a state) and a transition represents a change of state or the ending of activities that are input to the transition and the starting of activities that are output of a transition. One token in a place may indicate that the corresponding activity is being processed or that a resource is available in that state. The dynamic behavior of the system modeled by the Petri net can be captured by the transition firing rules. A transition is enabled to fire if there is at least one token in each place that is input to that transition. When an enabled transition fires, one token is consumed from each input place and a new token is sent to each output place.

With this basic model it is possible to represent precedence and synchronization dependencies between activities. It is also possible to have a hierarchical representation by expanding places (or transitions) into more detailed networks. A set of algorithms are available to analyze properties of the system modeled by the Petri net such as deadlocks, loops, reachability of a given state, etc.

Departing from the original formalism a large number of extensions, intended to simplify the models or to support the representation of additional attributes, have been proposed such as temporized Petri nets, colored Petri nets, predicate transition nets, etc.

The use of Petri nets as a basis for defining control algorithms / coordination plans led to the appearance of several high-level Petri net proposals, extending the basic formalism (Colombo, 1998). The invocation of a service application can be modeled as a side-effect of a transition (Camarinha-Matos and Osorio, 1990).

Although there is a wide acceptance of Petri nets for modeling manufacturing systems, its use in VE coordination is not a common practice. Some authors advocate its "superiority" compared to other formalisms due to its formal background, but its practical use in this domain is still quite limited, probably due to the existence of many variants of Petri nets and the lack of "standardized" software packages. One example of application of Petri nets in Virtual Enterprises for the agribusiness sector can be found in (Camarinha-Matos *et al.*, 1997a). In this example, different elements and relationships of the agribusiness supply chain have been considered. For instance, the model for the supplier of raw material, the

supplier of supplies and commodities for production, the client, as well as the producer-factory, supplier-factory, and the factory-client relationships.

Although Petri nets are not directly used, the basic concepts and mechanisms proposed by this formalism can be found in the background of many other tools used in workflow and business process management.

**Workflow management systems** represent a popular approach to manage and support business processes. A workflow consists in the automation of procedures where documents, information or tasks are passed between participants based on a predefined set of rules to accomplish an objective (Lawrence, 1997). Although not based on a sound formal background as Petri nets, workflow systems have two major advantages that justify their popularity:

- The architecture of a workflow system explicitly emphasizes the interoperation between heterogeneous legacy systems, taking care of the communication among those systems.

- The supporting technology (tools) has reached a quasi-standard status due to the joint efforts of the major software vendors through the Workflow Management Coalition (WfMC, 1994).

As a consequence there is a well-established market for this technology which justifies its adoption by the VE community, as it will be described below. On the other hand, the workflow research community is also overcoming a major drawback that is to cope with dynamic changes in workflow models. Good references can be found in the literature, such as (San-Yih et al, 1999) (Reichert 1998), and the German project "*Knowledge-Based Dynamic Modification of Workflows*", being developed at the University of Leipzig<sup>1</sup>, presents good solutions to that problem.

**Computer-Supported Cooperative Work (CSCW)** is the research field that examines the design, adoption, and use of tools (groupware) to support tasks that are carried out by groups of mutually dependent persons (Divitini and Farshchian, 1997). Coordination in CSCW addresses the problem of integration and harmonious adjustment of individual's work efforts toward the accomplishment of a common goal. According to (Brinck, 1998), the most common reasons people want to use groupware are: to facilitate communication, to facilitate group problem solving, to save time and cost in coordinating group work, etc.

Research activities on CSCW have originated various groupware tools, commercially available, in various categories, such as: computer conferencing, chat, application sharing, shared whiteboards, co-authoring, collaborative virtual environments, videoconferencing, etc. An extensive survey can be found in (Hofte, 1998). According to this author, and in spite of this variety, the offered support is still inflexible and lacking integration between different types of tools. A point to notice is that these tools were designed for human cooperation and many of them were not initially designed to strongly support autonomy and privacy among involved partners. Therefore, although there are some experiences of application of

---

<sup>1</sup> <http://www.informatik.uni-leipzig.de/ifi/abteilungen/db/Research/wf-modification.html>

such tools in VE environments (Bloch & Pigneur, 1995), they might present some drawbacks.

**Business Process modeling** is a well-established area in the enterprise world. The CIM-OSA (Open Systems Architecture for Computer Integrated Manufacturing) represented a milestone in the early attempts to model enterprises from multiple perspectives (AMICE, 1993). CIM-OSA and similar initiatives contributed to the consolidation of the enterprise modeling and engineering area. To model the activities in enterprises, CIM-OSA introduced the concept of business process (BP) that can be decomposed into a hierarchy of sub-business processes and enterprise activities (EA). The enterprise activities, that are supported by the so-called Implemented Functional Operations, represent the basic functional building blocks (library of services) an enterprise has to actually realize its processes. The behavior is represented by the Procedural Rule Sets (PRS) that identify the conditions under which the different activities will follow each other. The CIM-OSA enterprise Integrating Infrastructure is the mechanism that plays the coordination role by “executing” the coordination plan represented by the BP/EA/PRS. Strong similarities can be noticed between this approach and the workflow management philosophy. Current activities try to extend this modeling and coordination framework to the virtual enterprise domain (CIMOSA, 1998).

Although the conceptual contribution of CIM-OSA is considerable, the various “generations of CIM-OSA” projects sponsored by the European Esprit program were somehow disappointing in terms of realizations and support tools. One of the reasons was the approach of “re-inventing the wheel” initially followed by CIM-OSA that ignored all the complementary or competing activities. Only in a later stage a more open approach was adopted and then CIM-OSA results were used as a substantial input to a wider initiative promoted by IFIP and IFAC, the GERAM task force.

GERAM stands for Generalized Enterprise Reference Architecture and Methodology (IFIP-IFAC, 1997) and describes the methods, models and tools which are needed to build and maintain the integrated enterprise, be it a part of an enterprise, a single enterprise, or a network of enterprises (virtual enterprise or extended enterprise). GERAM is based on the works carried out by the AMICE Consortium on CIMOSA, by the GRAI Laboratory on the GRAI/GIM, and by the Purdue Consortium on PERA, as well as similar methodologies by others. The purpose is not to make yet-another-proposal for an enterprise reference architecture, but rather to organize existing enterprise integration knowledge. Therefore the business concepts introduced by those previous works were somehow combined by GERAM in a common framework.

Nowadays the BP modeling and analysis area represents a very active and growing market for which a large number of tools are available. Some examples are: ARIS form IDS Scheer<sup>2</sup>, Workflow modeler and analyzer by Metasoftware<sup>3</sup>, IsModeler by Modus Operandi<sup>4</sup>, Grade Modeler by the Grade Development Group<sup>5</sup>,

<sup>2</sup> <http://www.meridian-marketing.com/ARIS/index.html>

<sup>3</sup> <http://www.metasoftware.com>

<sup>4</sup> <http://www.intecsys.com>

FirstSTEP by Interfacing Technologies<sup>6</sup>, SIMPROCESS by CACI<sup>7</sup>, etc. An attempt to standardize the business process representation is given by the Process Interchange Format (PIF) initiative<sup>8</sup>. In general, these tools are based on graphical interfaces / graphical languages and allow for business process modeling, structure and communication modeling, simulation, and analysis: cost, information usage, performance (cycle times, errors), detection of bottlenecks, resource utilization, etc.

Apart from the different graphical look, the underlying mechanisms in these tools are inspired in the IDEF0, workflow, or Petri nets developments. However the main utilization of these tools is for modeling, simulation and analysis, not for the coordination itself and, so far, not easily usable in a VE coordination infrastructure.

**Coordination in MAS.** The area of Multi-Agent Systems (MAS), especially when involving Intelligent or Autonomous Agents, has also been discussing coordination issues and supporting mechanisms (Durfee &all, 1987).

The interaction capability, both among agents and between agents and their environment, is one of the basic characteristics of an agent. The definition of high level protocols – Agent Communication Languages (ACL) - to support these interactions has been a major work item. KQML (Finin et al., 1993) is one of the most widely known examples of such protocols. Another important mechanism coming from the early works on MAS is the contract net and negotiation protocols (Smith, 1980) that can be used for task assignment.

Applications of these initial concepts have been developed namely in the area of scheduling / agile scheduling (Rabelo and Camarinha-Matos, 1994). The extension of these concepts to virtual enterprises is the subject of the MASSYVE project (Rabelo et al., 1998). In fact a VE environment can be viewed as a MAS and VE members can be considered as agents or collections of agents that need to be coordinated in order to guarantee that the VE achieves its goal.

**Coordination languages.** The development of coordination languages constitutes a new field of study in programming and software engineering, particularly motivated by the needs of massively parallel and distributed systems. The starting point is the recognition of the need of programming models that explicitly deal with the concurrency of cooperation among large numbers of entities in massively parallel systems. Examples of such languages are LINDA, MANIFOLD, GAMMA, COOLL, PCL, etc. For a comparative survey of coordination languages see (Papadopoulos and Arbab, 1998). In spite of this growing activity, none of these languages has yet entered the enterprise modeling and VE application areas. One possible reason is that these languages have a complex syntax, more oriented to programmers / software developers than to business process modelers. The lack of simple graphical tools, although some initiatives are starting in this area, such as Visifold (Bouvry & Arbab, 1996), is another disadvantage of these languages when compared to

---

<sup>5</sup> <http://www.corpdyn.co.uk/Global.html>

<sup>6</sup> <http://www.interfacing.com>

<sup>7</sup> <http://www.caciasl.com/simprocess-product.html>

<sup>8</sup> <http://ccs.mit.edu/pif2.html>

alternative solutions found in the business process modeling area.

**Business-oriented mechanisms.** Other contributions for the identification of suitable coordination mechanisms can be found in some research activities that try to characterize and model the distributed business processes in a VE environment. One example in this direction is (Sonka, 1998) that proposes a model to describe market coordination mechanisms in the agribusiness supply chains.

### Workflow reference model

In an attempt to facilitate the interoperation among different components of a workflow system, the Workflow Management Coalition (WfMC) defined a reference model for workflow management systems. This model is composed of five key modules (Figure 1). For each module, there is a set of functions that should be accomplished by the commercial products covering that module. The WfMC is also defining the interfaces between the five modules.

All the workflow management systems have a number of generic components that interact using a set of predefined ways. Different products will offer different levels of capabilities within these generic components. To reach interoperability between the various products it is necessary to adopt a standard set of interfaces and data interchange formats in those components, which is the major contribution from the WfMC Reference Architecture.

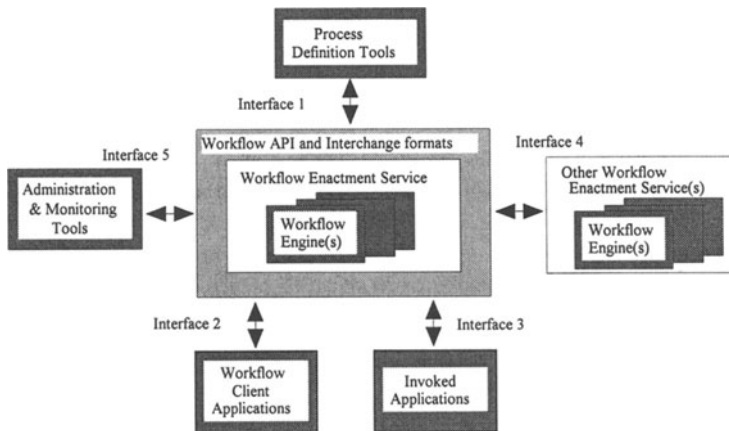


Figure 1 – Reference Model for Workflow Management Systems (WfMC, 1994)

A large number of projects in the VE domain have adopted the WfMC reference model as the basic model for implementing coordination. Examples are the projects NIIP (NIIP, 1996), VEGA (Zarli et al., 1997), PRODNET II (Camarinha-Matos and Lima, 1998), etc. Some reasons for the adoption of this model are the following:

- The WfMC is a well-known reference model towards the standardization in the workflow systems area;
- The WfMC includes a component to define properly a workflow model. Such



model can be used to flexibly support the configuration of the coordination plans;

- There is a formal language already developed to support the workflow model definition – the Workflow Process Definition Language (WPDL)(WFMC, 1997);
- The Workflow Engine is the executor of a workflow model. Thus, the coordination mechanisms in a VE can adopt some concepts from this component. I.e., the *VE Coordination Modules* can be viewed as specialized workflow engines executing a workflow model customized for a certain VE or company.

Although many projects have based their coordination mechanisms on the WfMC Reference Architecture, most of them have implemented their own workflow engines instead of adopting solutions from the market. This situation is justified by the need to have an open and flexible coordination mechanism during a phase in which the coordination requirements for a VE are evolving and, as such, not completely identified yet. Only a few of the tools offered in the market were designed as open systems. On the other hand, commercial packages are quite heavy due to many extra management functions that are not necessary in this context. In many cases these systems are oriented to office automation, although a few cases (e.g. Meteor, StaffWare) are designed as generic business process oriented tools. The development of a light coordination engine, able to be integrated with the other components of a VE infrastructure, is therefore a justifiable option.

### Communication and privacy

Communication mechanisms are an essential part of coordination to support exchange of information between interdependent activities / processes. In a distributed environment such as a VE the communication needs arise at two levels:

- a) Communications inside a VE member;
- b) Communications among VE members.

The processes / activities that take place in a VE member are supported by a number of tools, most of them heterogeneous legacy systems. The coordination system has to define an internal protocol to support the invocation of the services provided by those tools. From the activity coordination side, the “relevant data” structures from the WfMC model are adequate to support the data interactions among activities. The communications with legacy systems could be highly facilitated if some standard interface was available. In spite of many efforts, such as those of CIM-OSA and GERAM, there is no such standard. A more recent initiative launched by an association of major application developers – Open Applications Group (OAG) – that is trying to define specifications to enable business object integration across enterprise applications (OAG, 1997), represents a promising approach in this direction.

The communication requirements in terms of activities / processes that take place in different nodes of the VE are much more demanding. First of all, as open networks such as the Internet, subject to potential piracy acts, support communications, the issue of safety becomes of major importance. Furthermore, as the “executors” of the activities / processes are autonomous nodes that might have

some conflicting interests, additional requirements arise. Therefore, for inter-node cooperation activities, the following characteristics should be offered by the communication mechanisms:

- Confidentiality – preventing non-authorized “observers” to have access to the information being exchanged;
- Integrity – giving the partners the guarantee that what was received is what was sent;
- Authentication – giving the receiver the certainty about the identity of the sender;
- Non-reputability – allowing nodes to rely on “contracts”, i.e. giving them the confidence that partners cannot deny the commitments included in their messages.

A number of mechanisms have been proposed to support these requirements, particularly in the area of Electronic Commerce (Camarinha-Matos et al., 1997), (PETER, 1998), including cryptography, with symmetric and asymmetric ciphers, digital signatures, identity certificates issued by certification authorities, etc.

Besides the secure communications, another level of requirements is related to the semantics and representation of the exchanged data. For some classes of information, such as business-related transactions or technical product data exchange, international standardization activities led to standards such as EDIFACT or STEP. For other classes of information such as quality-related reports, manufacturing status follow up bulletins, business processes, etc., there are no established standards yet. Therefore, agreements have to be established among VE partners.

Finally, a related issue is the support for shared information. A member of the VE may decide to give access to its partners, with different levels of visibility, to part of the information it owns. The coordination mechanism has to support the secure access to this shared information by activities running in partner nodes.

From the topics discussed above, it may be concluded that a coordination mechanism for a VE infrastructure will be based on a kernel “triangle” composed by an activity / process – flow coordination engine, a secure communication infrastructure, and an information management system (Figure 2).

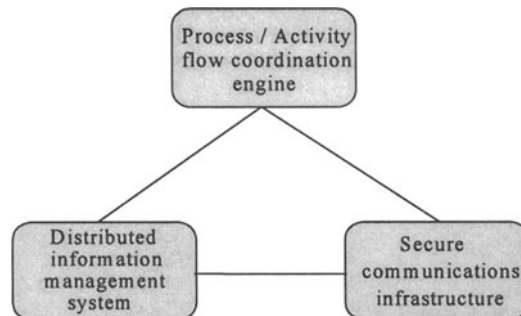


Figure 2 – Kernel components of a VE-coordination mechanism

## HIERARCHICAL COORDINATION NEEDS

### Motivation for hierarchical coordination

The coordination issues in a VE environment can be analyzed at different levels of abstraction, which may suggest a hierarchical approach to coordination.

Let us consider, for instance, a VE established to manufacture a bicycle. Different parts of the manufacturing process may be assigned to different members of the VE according to their core competencies. One node A becomes responsible for assembling all components in a final product, another node B is responsible for manufacturing the metallic parts, a node C supplies the plastic components, a node D produces moulds for plastic components, etc. Once a global business process is defined and scheduled and responsibilities are assigned to individual partners, it is obvious that the successful achievement of the common goal – delivery of the final product to the client – will depend on the proper and timely operation of each VE member. A delay in one node may jeopardize the common goal if appropriate corrective measures are not taken in time. Therefore, one enterprise (the assembler, for instance) may assume the role of VE coordinator and manage (supervise) the interdependencies among the various (distributed) business processes. A list of examples of supervision activities to be carried out by the VE coordinator can be found in (Klen et al., 1998).

On the other hand, zooming in inside each company, we can observe that each local business process may, on its turn, be decomposed into several sub-business processes whose activities are supported (performed) by (or with the help of) various applications / tools existing in the enterprise.

For instance, a “produce bike frame” process may involve:

*“Receive order from the coordinator”, “design frame” (using a CAD system); “generate process plan” (using a CAPP system), “plan production” (using a PPC system), “supervise production” (using a shop-floor control system); “send periodic progress reports to VE coordinator” (using the PPC system), etc.*

The interdependencies among these activities need to be properly managed.

This zooming in exercise can proceed and a simple activity such as “receive order from the coordinator” might imply a sequence of fine-grained activities. For instance:

*“Identify arriving message”, “parse order coded in EDIFACT format”, “temporarily store order data”, “ask advise from human operator on what to do with the order”, “notify PPC of the order arrival”, etc.*

Each of these steps may be supported by a different tool / service of the VE infrastructure. Instead of asking advise from a human operator, a company might want that arriving orders are immediately made available to the PPC system. Or this procedure could be applied to selected partners only, while for orders from other clients a manual step is necessary. Therefore, flexible coordination plans should be supported to allow for different behavior implementations.

**Re-engineering needs.** The implantation of the VE paradigm will require some re-

engineering and extensions to the legacy PPC and ERP systems. In a traditional sense these systems evolved as large monolithic applications, difficult to interface with. The VE coordination requirements will impose not only openness in terms of data accesses but also in terms of the control structures of these applications. The more traditional ones were designed to be driven by human operator interfaces and are not ready to react to events originated in the VE network.

When an enterprise decides to participate in a VE it implicitly gives up part of its autonomy that is transferred to the VE consortium, according to the established contracts. For instance, an enterprise might be obliged, by contract, to let its partners have access to some information on orders processing status, stock level, etc. This implies that some service functions from a PPC / ERP system must be invoked, in a controlled way, by a new coordination mechanism to satisfy the duties of the enterprise regarding its contract with the VE. In other words, a coordination layer, at the enterprise level, has to be superimposed on top of the enterprise applications.

One of the re-engineering approaches is to have a clear separation of the business-oriented processing functions (application services) from the application control structure. In this way a VE-oriented coordination mechanism, for instance based on a workflow-like system, can invoke the application services to fulfill the business processes needs. This approach is consistent with what the OAG calls "componentization", i.e. *"the process of breaking down business applications into functional components that have tangible points of integration to other components"* (OAG, 1997). Although OAG is motivated by application interoperability needs (to facilitate integration of components from different vendors) the approach also contributes to facilitate a flexible coordination philosophy, especially if the level of componentization reaches a fine grain level.

### **Coordination and contracts**

As contracts are the main vehicle that regulates the general execution of business processes between enterprises, they can be used as one way of coordinating the interdependencies among enterprises in a VE. A contract shall establish the conditions for the interdependence / cooperation among partners and, therefore, can be the basis for a coordination policy.

A VE coordination system could be based – at the higher levels – on clauses modeled from the business contract. For instance, if the contract states that partner A shall send to partner B, every Friday evening, a report on the production status regarding some distributed business process, two workflow plans can be defined, one for each partner, in order to automate the fulfillment of this clause. The workflow in partner A would include the activities necessary for gathering the production status data in the right time, formatting these data and sending the report to partner B. On the other hand, a workflow running in partner B would include the activities to supervise the arrival, in time, of the expected report and, in case it doesn't arrive, start some contingency procedure. A desirable extension to the PPC / ERP systems would be a workflow-based contract management system.

However, new cooperation forms require new contractual forms. Even for the

classical business transactions (like “placing an order”) there is no standard form of contract, i.e. each enterprise has a variant contract model. For more complex and uncertain forms of relationship such as those emerging for VE scenarios, there is a need to experiment and assess new contract forms.

Although in future some form of semi-automatic derivation of coordination rules from the contracts can be envisaged, in the current stage only a manual approach seems realistic.

## **COORDINATION AND CONFIGURATION**

### **Configuration needs and approach**

As discussed in previous sections, it is mandatory that the VE infrastructure and its coordination policies be flexible and configurable. Since every enterprise is different and wants to keep its autonomy, the infrastructure and coordination policies must be designed to be customized according to the decision of each company and the clauses of the cooperation contracts. For instance, as mentioned above, for one enterprise the sequence of steps required to process a specific incoming event from another enterprise, can significantly differ from the procedure defined for the same purpose in another company. Furthermore, there are a large number of factors (Camarinha-Matos and Lima, 1998), (Camarinha-Matos and Afsarmanesh, 1998) that impose a flexible approach. Examples of such factors are:

- Diversity of VE classes, in terms of duration, topology, coordination policy, visibility scope, etc.
- Diversity of roles played by each enterprise, such as VE member, coordinator, client, supplier.
- Diversity of VE coordination behaviors (trigger, broker and controller);
- Diversity of standards for information exchange; for instance, even when using EDIFACT, two partners need to agree on a common subset of messages and on the precise composition of each message.
- Diverse motivations for creating a VE: decomposition of large company into smaller units or aggregation of small firms into a larger (virtual) organization.
- Diversity of internal management policies, socio-organizational structures and computational resources found in each company, which also depend on the size of the companies.
- Diversity of rights and duties (e.g. to share and exchange the local information) that can be associated to each VE member.
- Diversity of contract / subcontract forms between members of the VE.
- Participation of a company in multiple VEs with different roles, rights and duties towards each VE.
- Evolution of support technologies, safety mechanisms and the legal framework for electronic commerce. All these areas are either showing a high dynamism or being in their infancy, which recommends a “least commitment” approach.
- Forms of interaction and cooperation between enterprises are likely to evolve with experience and trust building mechanisms. Based on acquired experiences it

is likely that new organizational paradigms and new contractual forms for cooperation among industrial companies will emerge.

From a coordination point of view, this situation strongly suggests that the coordination policies should not be implemented as “embedded code” in the VE infrastructure; they should rather be easily configurable for each particular context. Similarly, it should also be easy to change (reconfigure) these policies each time the environment or the contractual rules change.

One important aspect in the configuration process is its “efficiency” considered both in terms of the required time and background knowledge expected from the human configurator. Thinking in terms of business practice, it is expectable that an expert in business processes, not necessarily familiar with programming languages, will perform the configuration. Therefore, the definition of a coordination policy should not require strong programming skills. A graphical-oriented approach, such as the one suggested by some business process-modeling tools, seems to be an adequate approach.

### **Coordination and exception handling**

As a consequence of the complexity involved in the VE environments, the management of unexpected events and errors is a mandatory issue to be addressed by a VE infrastructure.

A good approach to introduce exception-handling mechanisms in such platforms is to separate the failure semantics from the program logic, which means separation of the exception detection from the exception handling. Exception handling supports not only the recovering of unexpected errors, but also the management of the predictable faults in the VE operation. Configurability is also an issue to be considered in exception-handling mechanisms. The exception-handling procedures could be, as much as possible, configurable being represented, for instance, by specific workflow plans.

## **CONCLUSIONS**

Configurable definition of coordination policies supported by a flexible coordination mechanism is a major issue in the design and development of a supporting infrastructure for virtual enterprises.

There is a growing interest for the definition of coordination mechanisms and languages in various disciplines, namely in various areas of computer science. At the current stage, the results from the workflow management and business process modeling areas are the ones mostly used by VE-related research projects.

The coordination issues can be discussed at various levels of abstraction and also encompassing the different roles an enterprise may play in a VE, which suggests a hierarchical coordination approach.

## Acknowledgements

This work was funded in part by the European Commission, Esprit program. The authors also thank the valuable contributions from their partners of the PRODNET II consortium: CSIN (P), HERTEN (BR), MIRALAGO (P), University of Amsterdam (NL), Universidade Federal de Santa Catarina (BR), Universidade Nova de Lisboa (P), Lichen Informatique (F), ProSTEP (Germany), Uninova (P), ESTEC (P).

## REFERENCES

1. AMICE (1993) CIM-OSA: Open Systems Architecture for CIM. 2nd revised and extended version, Springer-Verlag, Berlin.
2. Arbab, F. – What do you mean, Coordination?, Bulletin of the Dutch Association for Theoretical Computer Science, March 1998.
3. Block, M.; Pigneur, Y. - The extended enterprise: a descriptive framework, some enabling technologies and case studies in the Lotus Notes environment, Proceedings of the 2nd International Conference in Network Organisation Management, June 95.
4. Bond., A.; Gasser, L. - *An Analysis of Problems and Research in DAI*, Readings in Distributed Artificial Intelligence, pp.3-35, Morgan Kaufmann, 1988.
5. Brink, T. – Groupware, <http://www.usabilityfirst.com/groupware/>, 1998.
6. Camarinha-Matos, L.M.; Afsarmanesh, H. – Cooperative Systems challenges in Virtual Enterprises, *Proc. CESA'98 – IMACS/IEEE Multiconference on Computational Engineering in Systems Applications*, Vol. 3, pp. 403-408, Nabeul-Hammamet, Tunisia, 1-4 Abr 98.
7. Camarinha-Matos, L. M.; Carelli, R.; Pellicer, J.; Martín, M. - Towards the virtual enterprise in food industry, *Proc. ISIP'97, OE/IFIP/IEEE Int. Conf. on Integrated and Sustainable Industrial Production*, Chapman & Hall , Lisbon, Portugal, May 1997, ISBN 0-412-79950-2, pp. 73-84.
8. Camarinha-Matos, L.M.; Lima – *Configuration and coordination issues in a virtual enterprise environment*, Proceedings of PROLAMAT'98, Trento, Italy, 9-12 Sep 98.
9. Camarinha-Matos, L.M.; Lima, C.; Osorio, A.L. – The PRODNET platform for production planning and management in virtual enterprises, Proc. of ICE'97, Int. Conf. on Concurrent Enterprising, Nottingham, UK, Oct 1997.
10. Camarinha-Matos, L.M.; Osorio, A.L. - Monitoring and Error Recovery in Assembly Tasks, 23rd ISATA - *Int. Symp. on Automotive Technology and Automation*, Vienna, Austria, Nov. 90.
11. Carriero, N.; Gelertner, D. – Coordination languages and their significance, *Communications of the ACM*, Vol. 35 (2), 1992, pp. 97-107.
12. CIMOSA Association – Enterprise Engineering and Integration – Why and how, CIMOSA Association e.V., 1998.
13. Colombo, A. W. – Development and implementation of hierarchical control structures of flexible production systems using high-level Petri nets, PhD Thesis, Meisenbach Verlag Bamberg, Mar 1998, ISBN 3-87525-109-1.
14. Divitini, M.; Farshchian, B. A. - CSCW: Introduction and Bibliography, <http://www.idi.ntnu.no/~it-rbt/cscw/material/material.html>, 1997.
15. Durfee, E.; Lesser, V.; Corkill, D. - *Coherent Cooperation among Communication Problem Solvers*, IEEE Transactions on Computers, N 36, pp.1275-1291, 1987.
16. Finin, T. Weber, J.; Wiederhold, G. – Specification of the KQML Agent-Communication Language, The DARPA Knowledge Initiative External Interfaces Working Group, June 15, 1993.
17. Hofte, H. T. – Working apart together – foundations for groupware, Telematica Instituut, 1998, ISBN 90-75176-14-7.
18. IFIP-IFAC– GERAM: Generalised Enterprise Reference Architecture and Methodology , version 1.5, IFIP-IFAC Task Force on Enterprise Integration, Sep 1997.
19. Klen, A. A. P.; Spinosa, L.M.; Rabelo, R. J.; Ferreira, A. C. – Integrated logistics management support system: An advanced coordination functionality for the virtual environment, Proc. IMS'98 – 5<sup>th</sup> IFAC Workshop on Intelligent Manufacturing Systems, Gramado, Brazil, 9-11 Nov. 1998.

20. Krone, O.; Chantemargue, F.; Dagaëff, T.; Schmacher, M.; Hrsbrunner, B. – Coordinating Autonomous Entities, Proc. of SAC98, Symposium on Applied Computing, Atlanta, Georgia, USA, Feb 27 – Mar 1, 1998.
21. Lawrence, P. – The Workflow Handbook 1997, Wiley, Oct 1997, ISBN 0-471-96947-8.
22. Malone, T. W.; Crowston, K. – The interdisciplinary study of coordination, ACM Computing Surveys, Vol. 26 (1), 87-119, March 1994.
23. NIIP - The NIIP Reference Architecture, 1996, <http://www.niip.org>.
24. OAG – Open Applications Integration White Paper, Open applications Group, 1997.
25. Papadopoulos, G. A.; Arbab, F. – Coordination models and languages, in Advances in Computers, Academic Press, August 1998, Vol. 46: The Engineering of Large Systems.
26. PETER – Pilot for Electronic Marketing and Trading in European Small and Medium Enterprises, Technical Annex, Esprit 25326, 16 Dec 1998.
27. Rabelo, R.; Camarinha-Matos, L.M. - Negotiation in Multi-Agent based dynamic scheduling, *Int. Journal on Robotics and CIM (Pergamon)*, Vol. 11, N. 4, Dez 1994, pp. 303-309.
28. Rabelo, R.; Camarinha-Matos, L.M.; Afsarmanesh, H. – Multi-agent perspectives to agile scheduling, *Proc. Of BASYS'98, Intelligent Systems for Manufacturing* (Kluwer Academic), Prague, Czech Republic, August 98
29. Reichert, M.; Hensing, C.; Dadam, P.: *Supporting Adaptive Workflows in Advanced Application Environments. Proc. EDBT-Workshop on Workflow Management Systems*, Valencia, Spain, März 1998, S. pp. 100-109
30. Schmidt, K.; Simone, C. – Coordination mechanisms: towards a conceptual foundation of CSCW systems design, Computer Supported Cooperative Work: The Journal of Collaborative Computing, Vol. 5, n° 2-3, 1996, pp. 155-200.
31. Simone, C.; Schmidt, K.; Carstensen, P.; Divitini, M. – Ariadne: Towards a technology of coordination, Report of Riso National Laboratory, Riso-R-943 (EN), Nov. 1996.
32. Smith, R. – The Contract Net Protocol: High-Level Coordination and Control in a Distributed Problem Solver, IEEE Transactions on Computers, C-29(12) pp 1104-1113, 1980.
33. Sonka, S.T.; Cloutier, L. M. – System dynamics to evaluate information coordination in agricultural supply chains, Brazilian Journal of Agroinformatics, Vol. 1, N. 1, Sept. 1998.
34. VIVE – VIVE Reference model, 1999, <http://www.ceconsulting.it/VIVE/Results/default.html>.
35. P. Bouvry and F. Arbab, *Visifold: A visual environment for a coordination language*, In Coordination Languages and Models, vol. 1061 of Lecture Notes in Computer Science, pp 403-406, Springer-Verlag, April 1996.
36. WfMC - Workflow Management Coalition, The Workflow Reference Model - Document Number TC00 - 1003, Issue 1.1, Brussels, Nov 94.
37. WfMC - Workflow Management Coalition, Interface 1: Process Definition Interchange - Document Number TC - 1016, Issued on April 18, 1997.
38. Zarli, A.; Poyet, P. et al. - *Integrating Emerging IT paradigms for the Virtual Enterprise: the VEGA platform*. Proc. of ICE'97 4<sup>th</sup> Int. Conf. on Concurrent Engineering, Nottingham, 8-10 Oct. 97.
39. Hwang, S.; Ho, S.; Tang, J. - Mining Exception Instances to Facilitate Workflow Exception Handling, 6th Int. Conf. on DB Systems for Adv. Applications, 19-22 Apr 99, Hsinchu, Taiwan.