

MOBILE AGENT ASSISTED VALUE-ADDING COMMUNITIES FOR MASS CUSTOMISATION

NIKOLAOS V. KARADIMAS
VASSILI LOUMOS

*National Technical University of Athens
Multimedia Technology Laboratory
9 Heroon Polytechniou 9
Zografou Campus, 157 80 Athens, Greece
E-mail: NKaradim@central.ntua.gr
E-mail: loumos@cs.ntua.gr*

ALEXANDER TSIGKAS

*Production Management and Engineering
Democritus University of Thrace, Greece
Senior Consultant
Leonardo Group AG, Switzerland
E-mail: atsigkas@leonardo-group.com
Democritus University of Thrace, Greece*

Abstract: This paper proposes the design of an innovative management model for supply networks. This model promotes the requirement for close cooperation and adaptability as well as flexibility of ad-hoc structures throughout the supply network of custom made products and services. Structures of this type will respect principles that differ from the classical model of the industrial age, and evolve towards becoming self organising and self adapting entities. This type of structures will be organised, so that they provide flexible interfaces for plugging and adapting rather speedily into other ad-hoc cooperations for the purpose of providing custom made products and services. Once the purpose for which this “Virtual Integration” has been served, integration can be abandoned partially or completely and other structures can be formed in order to continue offering custom made and individual products or services in a new form.

Keywords: Supply Chain Management, Mobile Agents, Mass Customisation

INTRODUCTION

Custom made products and services require the design of Supply Networks that have special characteristics with respect to normal Supply Chains that serve more or less standard products and services. This research work examines the belief that in the future, markets will be more and more oriented towards individualised products and services that cannot be offered from one company only or even a number of them forming long term partnerships. These structures are too rigid for responding to the new challenges of the globalised markets and therefore new forms of organisation and management structures need to emerge. Furthermore, state-of-the-art web-based intelligent agents technology will be necessary to support operations of such management models.

This paper makes an effort to respond to this requirement by proposing a model for self-organising structures of ad-hoc cooperations for custom made products and services. Therefore, in this case, customisation addresses also the way individual professional teams should be structured, managed and evolved in order to form integrated systems whose purpose is to function as customised supply networks (CSN). These CSN are neither vertically integrated companies nor long term partnerships. On the contrary, they are ad-hoc

collaborations that have defined characteristics to serve the customisation and personalisation market of products or services.

This schema is described in the rest of the paper as follows: section 2 addresses further research and studies around the issues of how Supply Networks should be designed in order to take advantage of this new form of organisation and management structures to form CSN. In addition, section 3 describes briefly the principles for these ad-hoc structures. Section 4 introduces intelligent agents, a new breed of web-based technology with significant potential for a wide range of applications as well as a wider description of the mobile agents’ solution. Section 5 presents the business model proposed for the VAC accompanied by a reference on decomposition operators. Section 6 introduces a description of an intelligent agent application for manufacturing and the opportunities of agent deployment to illustrate the business case. Finally, in Section 7, conclusions and future developments are discussed.

VALUE ADDING COMMUNITY

A value adding community (VAC) (Tsigkas, 2005) is a supply chain of self-sufficient and autonomous organisation structures, which offer products or services. The composition of the VAC depends every time on the market needs and may change if

market needs change. Participants in the VAC may not be only enterprises but any entity that fulfils the interfacing requirements needed and described below. Because in this context, response time is the key issue, ad-hoc synergies can only be achieved if the participants of the VAC are structured and comply to certain agreed standards that are set and updated by the community. Therefore, for agile responses to the market needs it is necessary that VAC exhibits internally a cellular structure. These cellular structures share certain design characteristics that make the structure reusable, re-configurable and scalable. Rick Dove (2001) was the first to use these expressions in his remarkable book "Response Ability" in order to define the organisational structure of an enterprise he names "Agile Enterprise". These expressions are introduced here in order to define chains with configurable structures, which respond to the customisation and personalisation needs. Cells that need to cooperate and form each time a new VAC need to comply with certain organisational principles that make the ad-hoc communication possible with the least time and energy loss (Tsigkas, 2004). These principles form a certain framework that is known as the collaboration framework, which is accepted from the participants of VAC. This framework does not need to be negotiated and re-negotiated every time a new VAC or a chain of VAC needs to be formed. It is well known that cooperation and collaboration negotiations consume today a great deal of effort, time and energy on both sides in a bilateral relationship. In a multilateral relationship effort, time, energy and therefore cost grow exponentially. This reality makes collaboration agreements often practically impossible. The VAC framework, on the other hand, aims to alleviate this major constraint towards forming ad-hoc structures. These principles are stated in the next section, but not discussed in detail. It is proposed to the mass customisation community to initiate a discussion that should lead to the determination of a collaboration framework for the creation of VACs for custom made products and services.

PRINCIPLES FOR AD-HOC STRUCTURES

The principles are grouped into the following ten categories.

Self-Contained units: Components of a flexible structure are distinct, separable, self-sufficient units cooperating for a common purpose. An example of such an organisation is a unit of special forces in a mission, where every one knows the objective, his own role and the role of the others in the team. They do not sit down and interrogate each other to decide about individual trustworthiness and competency. The team leader may in fact choose the tactical

targets, but the task expert will decide how to accomplish the task. And any one of them could have been any one of many others.

Interfacing compatibility: Components of a flexible structure share defined interactions and interface standards and they are easily inserted or removed.

The individual candidate will have to fulfil certain standards with respect to the values that he or she will must have acquired in order to assure compatibility within a VAC. The problem with the management, in many cases, is not the standard, but the multiplicity of them. With respect to the replacement of a unit or component within a flexible organisation, what should be avoided is a total integration of the components or the units in the final scheme. The total integration of a member within a team may, on the one hand, have advantages in the operation of a team; however, it may also have negative effects when this unit or member has to be replaced. The objective here is to be able to speedily adjust the value creating process of the VAC in order to adapt quicker to market changing needs. For example, the integration of the suppliers in the production in the end of the 90s exhibited very good results when operated properly. When, on the other hand something went wrong, these suppliers were so closely connected to the system that replacement became extremely difficult.

Flat Interactions / Interactions without hierarchy: Components of a flexible structure communicate directly on a peer-to-peer relationship and parallel rather than sequential relationships are favoured. This is a special and very important feature for every supply chain. Gatekeepers that select, approve, censure or otherwise gate the communications of a system component are stealth members of the system (Dove 2001). For example, many businesses went through a strong process reengineering focus in the 1990s. A lot of that effort was directed at changing sequential processes into cell or parallel processes. In many areas of manufacturing, the concept of cellular processes took favour over sequential processes, partly because the processes that created the problem had the ability to attend to it immediately and directly.

Deferred Commitment: Components of a flexible structure are transient when possible, decisions and fixed bindings are postponed until necessary and relationships are scheduled and bound in real time. Deferred commitment basically keeps existing options open as long as possible. The underlying wisdom here is to not make a decision until it is absolutely necessary, knowing that more information arrives with time and decisions should take advantage of that flow.

Distributed control and Information: Components of a flexible structure are directed by objective rather than method. Decisions are made at point of maximum knowledge and information is associated locally, accessible globally and freely disseminated.

Empowerment at the team and the individual level is the classic example when people and organisations are involved. The use of this principle in the design of the VAC immensely facilitates system design since it decreases the need of communication between self sufficient, decentralised units to a minimum possible. Distributed information is often a reinforcement of the self contained unit principle.

Elastic Capacity: The number of components in a flexible structure may be increased and decreased widely within the existing framework.

This principle allows the flexible organisation like a VAC to increase and decrease capacity in order to adjust to the changing demand. For example, when outsourcing became a strategic discussion point in the early 1990s, the focus was generally on core competence issues. The strategy was to concentrate on strategic differentiations and to farm out the rest to others who would make a competency from what is considered to be supporting functions. That was the objective, but this concept was truly implemented only by companies that needed rapid capacity fluctuations. VACs should be capable of adjusting their response ability to the market by downsizing and upsizing as necessary to accommodate demand changes.

Redundancy and Diversity: Duplicate components are employed in flexible structures to provide capacity right-sizing options and fail-soft tolerance. Diversity among similar components employing different methods is exploited. This principle overthrows the approach of the single optimal solution, since it requires that sometimes getting a task done is preferable to relying on a single optimal approach. LSI Logic maintains a pool of resources capable of fabricating semiconductor wafers; some are wholly owned and some are qualified out sources. When an order arrives, a production chain can be assembled immediately from these pooled resources without waiting for some single resource or a resource with a unique capability that must finish its current commitment. Frequently it is more useful to have multipurpose people than dedicated experts.

Cross training in teams is often justified by the need to always have someone covering every need, regardless of who is absent for whatever reason. This obvious point is violated more often than not, especially in control cultures, where single-point approvals steal momentum, stop activities or put the

customer on hold while the gatekeeper is not available for whatever reason (i.e. vacation, out to lunch, ill e.t.c.). Collaborative learning and collaborative work have better results in a group with mixed backgrounds and mixed points of view.

Self-organisation: Components relationships in flexible structure are self-determined and components interaction is self-adjusting or negotiated. Self-organisation is typically related to natural systems such as eco-systems, societies and beehives, where seemingly intelligent or at least purposeful behaviour emerges from the total system though no central direction or control is evident. At the level of the VAC organisation, empowerment, teaming, listening to the voice of the customer, organisational learning and other concepts are extremely important. The principle of self-organisation basically means that the components of VAC have some discretion in deciding how to accomplish the goals established for them: what processes to employ, what priorities to set, and when to use which resources. In more advanced levels of VACs, the component can choose what to do, and with whom to do it.

Facilitated reuse: Components of a flexible structure are reusable or replicable and responsibilities for ready reuse/replication as well as for management, maintenance and upgrade of existing components are specifically designated. Can a duplicate of an existing component be readily created if another is needed? Can a necessary component be readily deployed when a new system must be constructed? Since within the VAC we are concerned within effective response to dynamic needs, someone or some mechanism is specifically charged with the responsibility for each activity involved. Attempts at formalised knowledge management are actually classical examples of facilitated reusability. However, reusability of knowledge is not sufficient. Knowledge, when created, should be made promptly available to all members of the VAC and to other VAC that may need it. This point is important since knowledge creation should be made with the least possible energy dissipation. The objective in the metacapitalistic market is to allow knowledge to achieve a high turnover in order to have real growth. It is therefore necessary to remove all barriers to the free dissemination of knowledge. Competitiveness will not be gained through knowledge retention. On the contrary, new knowledge should be quickly produced, disseminated and not accumulated and preserved as if it is a scarce resource (Tsigkas et al., 2004).

Evolving standards: Frameworks of flexible structures standardize inter-component communication and interaction, define component

compatibility and are monitored or updated to accommodate old, current and new components.

The purpose of the VAC framework is to facilitate reconfiguration, reuse and scalability. A framework should both constrain and enable these characteristics, bounding the set of potential configurations of an acceptable VAC system while encouraging full exploitation of the possibilities. It is the framework where all other principles described above find their right for existence and coexist in a dialectical interaction.

INTELLIGENT AGENTS

Traditional cooperations activities require a large effort from the participants collecting and interpreting information on products and/or services, making optimal decisions and finally entering appropriate transaction information. This difficulty can be alleviated through the use of intelligent agents.

Intelligent agents help to automate a variety of these cooperation activities, mostly time consuming ones, and thus lower the transaction costs. Intelligent agents differ from “traditional” software in that they are personalized, social, continuously running and semi-autonomous. In this way, the cooperation between the members of a VAC becomes more user-friendly, semi-intelligent and human-like.

These qualities are conducive for optimizing the whole exchanging experience and revolutionizing supply networks, as we know them today.

An agent is a software entity which functions continuously and autonomously in a particular environment often inhabited by other agents and processes. The requirement for continuity and autonomy derives from the human desire that an agent should be able to perform activities in a flexible and intelligent manner responsive to changes in the environment without constant human supervision. An agent that functions over a long period of time should also be able to learn from its experience. Furthermore, an agent that inhabits in an environment with other agents and processes, it should be able to communicate and cooperate with them, and perhaps move from one place to another in doing so, as well.

Mobile Agents

According to Nwana (1996) mobile agents are computational software processes capable of roaming wide area networks (i.e. WWW), interacting with foreign hosts, gathering information on behalf of its owner and coming back home having performed the duties set by its user. These duties may range from a hotel reservation to

managing a suppliers’ network. They are autonomous and they cooperate, but differently from collaborative agents. A mobile agent can exchange data or information with other agents without necessarily giving all its information away.

The Mobile Agents’ Solution

The mobile agent concept is illustrated in Figure 1. A client computer (contractor or Provider) consists of an application environment, for example Microsoft Windows, which contains one or more applications for interaction with the VAC server. These applications may include information searching and retrieval, transaction front-ends, etc.

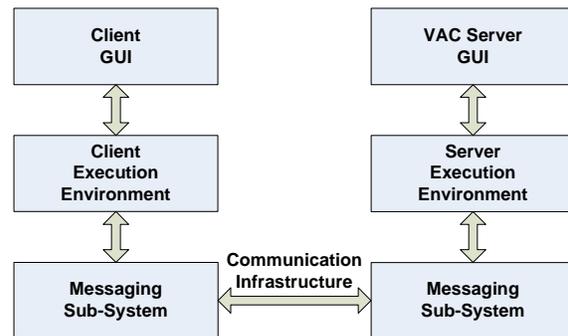


Figure 1: The Mobile Agent conceptual model

These applications are bound to an execution environment for mobile agents. Via the Graphical User Interfaces (GUIs), the application can pass parameters to various classes of agents programs, and likewise the agent programs can return parameters to the application programs. These classes may be part of the basic agent execution environment, agents distributed with Windows applications or agents received by the client from the server or other peer on the network. On the other hand, there may be no application program, but the agent programs could themselves perform presentation on the client device’s GUI and collect information directly from a keyboard or other input device. In this case the agent programs - or the agent execution environment - must bind to the user interface libraries of the client device.

The agent execution environment will also need to bind to the message transport service in order to send and receive mobile agents via the communication infrastructure.

When an application needs to perform a transaction, it will assemble the required information and then pass this via the GUI into the agent execution environment.

This will initiate the execution of an instance of a particular class of agent as a process within the

agent execution environment. This may correspond to an operating system process or an operating system thread or it may be managed by a threads package within the agent execution environment. The agent execution environment will have access to many different agent programs, which provide different services to the client applications. For example, one may act as a delivery agent for a database retrieval/request to the VAC server, submit the request and return the result to the client application, another may classify the different solutions/choices that the VAC server offers, yet another may navigate its way among multiple servers (in the case where there are more than one) asking each in turn for updates on a particular product or service.

Preferably the agent is expressed in an interpreted language in order to support heterogeneity. Of course, there is a performance penalty for this, but since most of the agent processing is done not in the agent itself, but rather in the functions to which it binds, this may be acceptable. Interpreted languages also have the advantage of late binding; this enables the agent to contain references to functions or classes not present on the system where it is launched, but which are available at its destination. Interpreted languages are also easier to render secure, since the language developer explicitly controls what system resources are accessible.

The information assembled by the application is accepted by the agent as part of its initialization and at a certain point in its execution, the agent will execute an instruction which either the current agent process is suspended in the agent execution environment, or a new agent process is created. Then the suspended process or the new process, including its process state, stack, heap and all external references are collected and processed into a message expressed in a machine independent form.

This step is facilitated if the agent is built in an interpreted language. After that, the message is handed to the message sub-system and routed directly or indirectly to the destination VAC server, where it is delivered by the server's message sub-system to the agent execution environment. Next, in the agent execution environment the received message is reconstituted into the executable and the process or thread is dispatched. Finally, execution continues at the next instruction in the agent program [Harrison et al., 1995].

During execution at the VAC server, the agent passes the information it received from the client application to VAC server application functions and perhaps receives other information in return. At the completion of this stage, it might terminate its

execution or it might simply suspend at the server, waiting for some event to be delivered from a server application or it might repeat the migration progress, either by forking a new process or by suspending and migrating itself [Harrison et al., 1995].

Personalization of Server's behaviour

Mobile agents also enable the contractors and the vendors to personalize the VAC Server behaviour. The VAC server should offer basic GUIs and export them via the bindings of the agent execution environment for exploitation by mobile agent programs.

Of course, the user (contractor or vendor) has then the freedom to use the VAC server as he or she desires. Thus, in the example discussed below, if any contractor wishes to browse the catalogues of several vendors instead of simply using the client application provided by the vendor, he or she has the freedom to dispatch a mobile agent to forage the vendors' servers for information relevant to a product or service.

On the other hand, the drawback of this facility is the equal feature with which viruses and so forth could be installed into to VAC Server.

THE VAC BUSINESS AND DATA MODEL

In this model (Figure 2) it is assumed that the VAC participants are already qualified according to the standards established based on the previously stated principles and that their profiles are managed via a VAC Server.

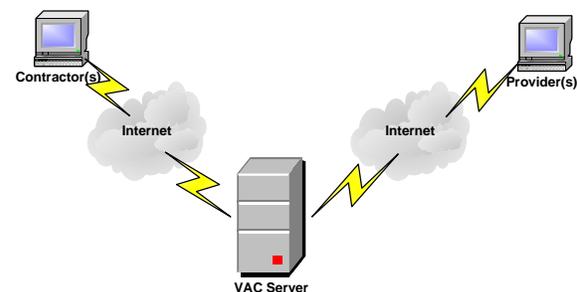


Figure 2: Contractor-to-Provider communication.

The typical business case (Figure 3) starts with the request of a member of the VAC (thereafter the Contractor) for the provision of a product and/or service looking for potential providers in the community (thereafter the provider) that may fulfill his/her request.

This problem may be approached by answering the following questions:

1. How should the request be modeled?

2. What should the content of the model be?

To answer both questions a generic way is chosen. A request can be modeled as a hierarchical structure for grouping the standards necessary to be satisfied in all stages of the life cycle of a product or a service to be provided. As a convention these groups in the hierarchical structure are named: decomposition operators, a term borrowed from the STEP Project [Deshpande and Nazemetz, 1999]. The content and the sequence in which these standards should be applied will then depend strongly on the individual product or service requested. Therefore the request from contractor should satisfy basically two requirements:

1. The decomposition operators required with the standards to be followed
2. The sequence in which these operators should be applied

The following list of decomposition operators, while extensive is not an exhaustive list. Other operators may be needed for particular design/decisions not envisioned by the authors. The list should, however, provide insight into the more common decomposition operators [Deshpande and Nazemetz, 1999].

Decomposition Operators

1. **Organizational:** This decomposition operator groups the standards according to the organization that is responsible or sanctions the standard. Some of the examples of such organizations are ISO, IEC, IEEE, ANSI, DIN, and ASTM.
2. **Geographic:** This decomposition operator groups the standards according to geographic range of the standards application/product market. It could be international, national or regional standard. For example, the use of DIN (German) material standard may be applicable only in Germany or where its exceeds the requirements mandated by other local standards with which it is being harmonized.
3. **Industry:** This decomposition operator groups the standards according to the industry in which the activity takes place. The standards applicable for industries are often different from the standards applicable to the same function in other industries. Examples of industrial standards are the differences between the standards for pipe and tubing in residential housing and the standards for pipe and tubing in chemical processing or shipbuilding.
4. **Type or Class:** This decomposition operator groups the standards according to the type or class of industry to which they apply. A standard can be used for defense/non-defense application, commercial/military, and private/public (e.g. handicap accessibility standards are different for public and private buildings).
5. **Industrial Sector:** This decomposition operator groups the standards according to industrial sector. Some of the examples of these sectors are agricultural, extraction, manufacturing, and service.
6. **Functional:** This decomposition operator groups the standards according to its functional application. Some of the examples of this are ISO 9000 and ISO 14000 series standards, which apply to managerial functions and ISO 10303, which applies to data exchange function.
7. **Product Type:** This decomposition operator groups the applicable standard for a particular product. A particular product, for example an aircraft, will be affected by standards from many sectors like military standards, some proprietary manufacturers standards, and some

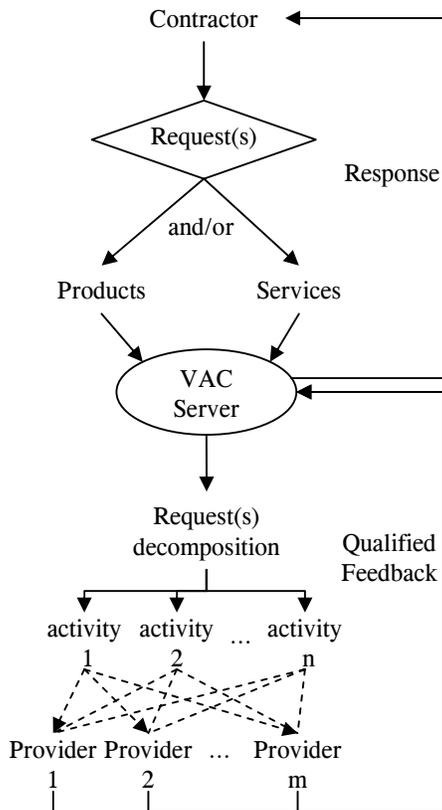


Figure 3: The communication procedure.

mandatory standards like Federal Aviation Administration (FAA) standards.

8. **Product Stage:** This decomposition operator groups the applicable standards for different stages of product life cycle. The different stages of product life cycle are identification of need, design, prototyping, manufacturing, service, and disposal.
9. **Viewpoint:** This decomposition operator groups the standards according to the observer’s viewpoint. A user of standard could be have a specific use (user) viewpoint or can have a harmonization or development (developer) viewpoint.
10. **Organization type:** This decomposition operator groups the standards according to the organization type. Examples would include Original Equipment Manufacturers (OEM), Small and Medium Enterprises (SME), Public and Private Industries.
11. **Conformance Requirement:** This decomposition operator groups the standards according to the conformance requirements. A standard, for example, could be mandatory or not.

To model the structure of the decomposition operators as well as the sequence in which these should be applied, the tool of Design Structure Matrix (DSM) can be used. The contractor will have to construct the DSM for his/her request and send it to the VAC server.

Table 1: A master matrix with the decomposition operators.

	decomposition operator	Standard No							
1	Organizational:	1	2	3	4	5	6	7	8
2	Geographic	1	2	3	4	5			
3	Industry	1	2	3	4	5	6	7	8
4	Type or Class	1	2	3	4	5	6		
5	Industrial Sector	1	2	3	4	5	6	7	8
6	Functional	1	2	3	4	5	6	7	8
7	Product Type	1	2	3	4				
8	Product Stage	1	2	3	4	5	6	7	8
9	Organization type	1	2	3					
10	Conformance Requirement	1	2	3	4	5	6	7	8

The structure of DSM is suitable to represent both requirements: the decomposition operators required with the standards to be followed and the sequence in which these operators should be applied. This is illustrated in the following example:

Step 1: In the first step the contractor must construct an individual matrix with the decomposition

operators that the provider should know for delivering the product or service.

This matrix can be realized by using the decomposing operator methodology or tool that STEP provides or any other method that he/she finds suitable to the purpose. The choice should be made out of a master matrix that includes all possible standards (Table 1).

Table 2: The customized matrix to serve a specific purpose.

	decomposition operator	Standard No							
1	Organizational:	1							
2	Geographic		3						
3	Industry						6		
4	Type or Class								
5	Industrial Sector				4				
6	Functional								8
7	Product Type	1							
8	Product Stage			4					
9	Organization type		2						
10	Conformance Requirement								7

It is obvious that not all decomposition operators may be applicable, as in this case Type or Class (4) is not required (Table 2).

Step 2: In the second step the contractor sets up a DSM (Table 3) in order to specify the order in which these standards should be applied. The “X” in the matrix can be read from vertical to horizontal in order to denote the sequence in which these standards should be applied: in this example, number 7 is applied before number 2, number 8 is applied before number 5 and so on.

This final DSM is then sent to the VAC server. The server forwards these elements to the VAC network for searching and finding possible partners that can fulfill the requirements originally requested by the Contractor.

Table 3: The Design Structure Matrix (DSM)

	decomposition operator	1	2	3	4	5	6	7	8	9	10
1	Organizational:										
2	Geographic	X						X			
3	Industry				X						
4	Type or Class										
5	Industrial Sector								X		
6	Functional		X								
7	Product Type										
8	Product Stage										X
9	Organization type				X						
10	Conformance Requirement										

The server waits for the response from the network in order to qualify the answer based upon compatible combinations with respect to the request and feeds it back to the Contractor. The Contractor

can then decide on the available options and in this case move on to update the formal agreement already existing among the members of the ad-hoc partnership to substantiate the cooperation. The agreed service can then be performed and the work will be evaluated upon its completion.

Evaluation data will become immediately available at the VAC server updating the profile of the ad-hoc participants. Evaluation concerns all members of the ad-hoc partnerships, contractor(s) as well as provider(s). Setting up the request, deconstructing it into individual activities, forwarding and feedback of qualified responses are processes undertaken by intelligent agents. The generation of compatible combinations is performed through specially developed algorithms that run at the level of the VAC server.

AGENT APPLICATION IN VAC FOR MANUFACTURING

In order to provide a business case of how and where agent-based computing can improve the cooperation within a VAC, a manufacturing scenario is offered, where agents can be effectively deployed, and operations such as procurement and supply-chain management, are identified.

A prototype multi-agent system to facilitate a bidding process for printed circuit board/assembly is discussed in some depth, identifying some of the functions that should appear in that kind of process. In particular, this prototype includes a specialized agent capable of retrieving technical design specification information stored in a standard file format. This could be a Computer-Aided Design file specification language standardized from the community itself and following an approved number of quality standards.

Defining quality is not an easy task. All participants have their own definitions of "quality" based on their subjective experiences. In order to have an objective set of guidelines that both customers and suppliers can refer to, the community should develop a set of quality standards to provide benchmarks for a participant's workmanship and the materials to be used.

A Manufacturing Project

A manufacturing framework develops a process-flow model, which identifies those internet-facilitated transactions for printed circuit board design, fabrication and assembly. The model identifies the major functions in the printed circuit assembly production process. There are certain functions that should appear in the aforementioned scenario.

Firstly, the original equipment manufacturer (OEM) needs to procure the boards and to generate a request for quotes. Therefore, the manufacturer (contractor) ships the request to a list of electronics manufacturing vendors, which should be approved members of the VAC. Next, the VAC Server may – if necessary- decompose the request to a number of activities and then each sub-request is forwarded to a number of relevant and available providers.

After that, the VAC Server receives the responses from each available provider and generates a list. This list may include information on availability indicators, cost and any other data, based on the particular request of the contractor. Depending on contractor's request, the list may return prioritized or in any other form desired.

The manufacturer and the vendors often conduct negotiations of terms. Then, if the manufacturer receives several quotes, he evaluates them and finally selects the winning bid. When the printed circuit assembler receives the order and the product definition package, it will often perform a manufacturability test and ship the report back to the manufacturer. On the other hand, engineering change order often occurs and needs to be evaluated and authorized if warranted, to be processed as an order for a new printed circuit assembly.

Opportunities for Intelligent Agent Deployment

Agent technology can assist in performing a variety of tasks. Agent research communities are building intelligent agents for performing negotiation, mediation, and brokering services. Agent technology also encompasses expert systems approaches that can perform complex scheduling, monitoring and alerting type of services. There are many potential areas for agent application in manufacturing services, such as:

Flexible Customer-to-Supplier Interfaces: Points at the opportunity for agent approaches to 'wrap around' or completely circumvent the existing form-based interfaces on the Web that have pre-defined syntax, implicit semantics, unpublished interaction protocols and, instead, enable automated, on-demand Customer-to-Supplier interface construction.

Optimized Negotiation of Service Cost and Terms: Currently, a human expert is solely responsible for negotiating an 'optimal' set of terms and the cost of service. Often, this person has a limited view of the business situation and cannot react immediately to a new business state of affairs. Agent approaches carry the promise of embedding significant decision making capabilities.

Efficient Intra-Enterprise Technology Adoption and Adaptation: Few means currently exist to accelerate adoption and adaptation of new e-commerce technologies within an enterprise. On the contrary, an agent should provide easier integration with legacy systems through usage of shared languages, ontologies and efficient updates of interaction protocols.

Efficient Engineering Change Order Processing: Each participant in the engineering and manufacturing process should manually sign-off on the change and make sure that the change is appropriately reflected in the part of the process for which the participant is responsible. Intelligent agents have the potential to make this process much less tedious and error-prone.

Efficient Inter-Enterprise Interaction Technology Support: Very few means currently exist to accelerate the adoption of new Customer-to-Supplier communication technologies across enterprises so that the enterprises can quickly engage in new inter-enterprise interactions.

CONCLUSIONS AND FURTHER RESEARCH

An attempt has been made to define the context in which future supply chain networks, for offering custom made products and services, should operate. In this context intelligent agents have been presented and discussed as being the vehicle for offering ad-hoc variety in the constitution of Value-Adding-Communities (VAC). Such communities need to be organised respecting basic principles. These principles must be defined in detail in the future so that further research and development can take place in developing special intelligent agents for the fulfilment of these principles. Furthermore, the introduction of intelligent agents in the set up of ad-hoc partnerships within a VAC can induce a breakthrough leading to a drastic reduction in time and effort requirements that traditional ways of setting up partnerships are experiencing today. As a first step, it is essential to develop the necessary agents that can undertake the tasks described above. Then, it is suggested that VAC members can periodically update the VAC server for any changes in their profile of offered products or services and on their capabilities in general. In this way, a search for compatible partners can be even further accelerated.

REFERENCES

Deshpande S. and Nazemetz J.W., 1999. Decomposition Methodology for Making the Standards Problem Tractable http://www.okstate.edu/ind-engr/step/WEBFILES/Papers/Decomp_Meth_index.html
Design Structure Matrix: <http://www.dsmweb.org/>

Dove R., 2001. "Response Ability – The language Structure and Culture of the Agile Enterprise". John Wiley and Sons Inc.
Harrison C.G., Chess D.M., Kershenbaum A., 1995. "Mobile Agents: Are they a good idea?". IBM Research Report, T.J.Watson Research Center, NY.
Karadimas N.V., Loumos V. and Tsigkas A. (2006): "Self Organising Structures of AD-HOC Cooperations for Customized Products and Services". In Borutzky W. Orsoni A. and Zobel R. (eds.): *ECMS 2006 – Proceedings of the 20th European Conference on Modelling and Simulation*. Bonn, Sankt Augustin, Germany, pp. 321-326.
Nwana H. S., 1996. "Software Agents: An Overview", *Knowledge Engineering Review*, 11, No.3, pp. 1-40.
Tsigkas A., 2004, "For a new society of citizen: the state, the citizen, the market and the value of knowledge in the MetaCapitalism", (unpublished book written in Greek).
Tsigkas A., 2005, "Mass Customization through Value-Adding Communities". *3rd Interdisciplinary World Congress on Mass Customization and Personalization*, Hong-Kong.
Tsigkas A., Papantoniou A. and Vassili Loumos, 2003. "Towards Knowledge Based Mass Customization: a Framework for an Demand Design (ODD)". *2nd Interdisciplinary World Congress on Mass Customization and Personalization*, TU Munich.

AUTHOR BIOGRAPHIES



NIKOLAOS V. KARADIMAS was born in Athens, Greece and he graduated from Patra T.E.I. in 1995 with a Bachelor's degree in Electrical Engineering. He then received a Bachelor's degree with Honours in Electronic Engineering and a Masters degree in Computer Science from Glasgow Caledonian University, Scotland in 1997 and 1998, respectively. He also received a Masters degree in Distributed and Multimedia Information Systems from Heriot-Watt University, Scotland in 1999. Since 2002 he is a PhD candidate in National Technical University of Athens. Furthermore, since 2001 he is teaching Informatics in Hellenic Army Academy and New York College, since 2003 he is teaching in T.E.I. of Chalkida and since 2005 he is teaching in Technical NCO Academy, as well. He is a member of the Greek Chamber of Engineers, member of IEEE and member of IEE. His research interests are in the fields of Databases, Optimization Techniques, Geographical Information Systems, Information Retrieval, Decision Support Systems and Multimedia.



VASSILI LOUMOS received the Diploma of Electrical Engineering and the PhD degree in Computer Science from the National Technical University of Athens (NTUA), Greece in 1975 and 1989. From 1975 to 1978 he was a research engineer at the

Centre National de Recherche Scientifique, Paris, France. From 1980 to 1982 he was with CGEE ALSHTOM as a field engineer. In 1990 he joined the faculty of the School of Electrical & Computer Engineering of the NTUA where he is presently a professor. He is teaching Multimedia Technologies and Computer Graphics. His research interests are in the fields of Multimedia, Computer Vision and Internet Navigation.



ALEXANDER TSIGKAS received a BSc in Physics from the University of Athens in 1974, a MSc in 1975 from the department of Electrical Engineering and a PhD degree from the department of Mechanical Engineering in 1978, Salford University in the United Kingdom. He is an active member in the community for Mass Customisation, being a speaker in three World Congresses. He works as a senior consultant for the Leonardo Group and he is a senior lecturer at the Democritean University of Thrace, Greece in the School for Production Engineering and Management. His experience expands to over 20 years in Industry with focus on Business Process Reengineering, especially where, effective and efficient Supply Chains is a key factor for the success of the customer. In the last years 10 years helps Enterprises in Central and East Europe, the middle and far East, who have chosen Flow or lean methodologies in order to implement Supply Chain strategies in their effort to increase competitiveness and leadership in their markets.