ON EMERGENCE OF VIRTUAL ORGANIZATIONS: A MULTI-AGENT PROBLEM-SOLVING APPROACH

Mihaela Ulieru
Electrical & Computer Engineering Department
The University of Calgary
Calgary, T2N 1N4, Alberta, Canada
ulieru@ucalgary.ca
http://isg.enme.ucalgary.ca

Rainer Unland
Institute for Computer Science
University of Essen
45117 Essen, Germany
Unlandr@cs.uni-essen.de
http://www.cs.uni-essen.de/dawis/

Abstract. An approach to the emergence of Web-centric virtual organizations is introduced based on a contract-net approach extended by cooperative problem-solving. The paper presents the key concepts of this emergence model that uses latest results obtained by the Foundation for Intelligent Physical Agents (FIPA) in developing new standards for platform interoperability, to enable the development of global collaborative e-Applications. As main engine for the emergent properties tying together the virtual organization we propose an 'in-house' solution which we refer to as MAPS (Multi-Agent Problem Solving) which allows agents to cluster and collaborate in a way that fits best to the individual goals of each agent (e.g .by maximizing a profit function).

1 Introduction

Complex information systems (IS) in traditional, i.e. functionally structured, organizations most likely encounter systems that were in principle constructed in a purpose-driven top-down way. Consequently, the requirements and aims of the underlying organization have both explicitly and implicitly influenced the design of these systems respectively of their components. As long as these requirements can be assumed to be relatively stable this is a feasible approach to system design, since changes occur seldom and thus can be accounted for by reengineering processes.
Opposed to this stability assumption are modern forms of organizations that have emerged recently. For example, *process orientation* and *virtual organizations* (VOs) promise more flexibility, adaptability and dynamicity. A virtual organization is a *temporary alliance* of enterprises that *cooperatively* work together to *share skills* or *core competencies* and *resources* in order to better respond to *business opportunities*, and whose cooperation relies on *computer networks* and a *cooperative*, yet *distributed information systems structure*. *Emergent* virtual organizations go one step further by constantly monitoring their performance and the market in order to improve their overall performance and efficiency, i.e., they permanently check whether there are (more) suitable possible partners available on the market which may either replace existing ones or add to the overall aim of the virtual organizations in a positive way. This feature may result in frequent organizational changes that have to be reflected by corresponding changes to the underlying information system (architecture). Another organizational structure that is discussed more recently are *supply webs* (see, e.g., [Lase98], [FiFR02]). Nowadays, a supply chain is less a chain than a complex web of intersecting supply chains. Even from the perspective of one big company, a highly impressive number of nodes will make up this supply web, since a huge number of companies may deliver parts for a product, with many products being produced by the company in question. Partnerships between autonomous business entities in these supply webs can be flexibly contracted or withdrawn and are predominantly short-dated. This may cause complex coordination problems since the resulting many-to-many interactions and instantiated supply paths are not stable but may dynamically change. In this paper we will address such kinds of organizations and will show how the problem of emergence, dynamicity, flexibility, stability, robustness, and adaptability can be dealt with.
The above characteristics strongly imply the use of a multi-agent system (MAS) architecture on the implementation level in which organizations are represented/cloned by one or more agent(s) at their interface. In a MAS environment it is assumed that every agent is autonomous and that the underlying agents architecture is in principle flat, i.e., agents are neither sub- nor superordinated. This is in congruence with the formation process of a virtual organization on the inter-enterprise level as it is usually discussed in literature (*horizontal* integration). However, in contrast to this one level view we argue that the above demands can only be achieved if the involved member organizations are deeply integrated and not only on the surface (*vertical* integration).

Since a virtual organization is only a temporary conglomerate that is established to quickly react to complex demands of the market it is mandatory that every decision on the inter-enterprise level is propagated and reflected in each member organizations through all levels up to the deepest level, the machine respectively cell level. This view extends the one level view to a hierarchical view. In order to integrate such a hierarchical structure in a society of autonomous agents, we advocate the concept of *holonic MAS*. It consists of several layers each of which represented by recursively nested self-similar structures (so called holons) which dynamically adapt themselves to achieve the design goals of the system. In holonic MAS these holons are groups of agents. In contrast to a normal MAS, agents that form a holon, need to accept a (partial) loss of their autonomy. However, they do not need to waive their autonomy completely. They can leave a holon and act autonomously or rearrange themselves as new holons. According to this view a holonic agent consists of sub-agents, which can separate and rearrange themselves and which may themselves be holons. Besides the introduction of a holonic MAS architecture for emergent virtual organizations we will show how this architecture can be implemented on top of the FIPA architecture. The Foundation for Intelligent Physical Agents (FIPA) is a non-profit organization that deals with standardization issues with respect to agents and MAS.

In the next section (section 2) we will first introduce the basics, namely virtual organizations, the demands they place on appropriate information systems, their extension in the direction of holonic enterprises, the notion of emergence and patterns of virtual collaborations. In section 3 we will discuss how our approach to emergent virtual organizations (emergent holonic enterprises) can be implemented using and extending the FIPA architecture as an enabling technology. We especially will discuss how the recently proposed standards for web-services and web service flow languages can contribute to the formation and performance of virtual organizations. Agents act on behalf of their organization. Therefore, they need to represent and follow the objectives of the organization they represent on the one hand and the objectives and aims of the virtual organization on the other hand. In section 4 we will discuss these different kinds of goals and how these goals can be expressed and managed in agents in a coherent and harmonious way. Finally, section 5 will conclude the paper.

2 Foundations and State-of-the-Art

In this section we will introduce the basic concepts and discuss the state-of-the-art. Starting with an introduction of the concept of *virtual organizations* we will extend this “single-layer” architecture to a multi-layered architecture by introducing the term *holonic enterprise*. Holonic enterprises form the basis for our proposal of *emergence* in *holonic enterprises* that we will introduce afterwards. Finally, we will discuss patterns of virtual collaborations.

2.1 Basic Characteristics of Virtual Organizations

Today’s organizations – to be successful – need to be highly flexible and adaptable. This insight has let to the introduction and discussion of *virtual organization*. A first pretty abstract definition is that it is a *temporary alliance* of enterprises that *cooperatively* work together to *share skills* or *core competencies* and *resources* in order to better respond to *business opportunities*, and whose cooperation relies on *computer networks* and a *cooperative*, yet *distributed information systems structure*. Up to now a commonly agreed on more precise definition is still lacking. However, the following four features are commonly accepted as mandatory characteristics:

1. Purpose-driven

Virtual organizations are formed for specific reasons and with clear overall common goals and objectives in mind. All members have agreed on these goals and objectives.

2. Flexible organizational structure

The organizational structure of a virtual organization can be quite dynamic; i.e. can change numerous times during the life time of the virtual organization. Davidow and Malone ([DaMa92]) distinguish between the *inside* and the *outside* view of a virtual organization. From the outside the shape of a virtual organization may change continuously. It may extend if additional tasks are identified which cannot be covered by existing members or if an external company seems to be a useful supplement for the virtual organization. Companies may leave because their internal goals do no longer meet the goals of the virtual organization. Finally, companies may have to leave if their value for the virtual organization does not meet the expectations. In the inside we have to deal with a permanent demand driven restructuring process with respect to functional units like groups or departments.

3. Autonomy of members

Members join a virtual organization intentionally. This implies that they are and will remain autonomous and independent. This, however, may result in that individual organizations may be driven by diverging or even conflicting aims. With respect to the underlying IS it means that each member keeps the responsibility for and the control of information concerning its part in the virtual organization. This can only be realized respectively ensured by a decentralized design for the information system of a virtual organization. It is not possible to integrate the individual system components of its members into one monolithic whole but they need to be kept separate and under the respective control of the underlying member organization.

4. Temporal membership

A virtual organization is a temporary network of independent organizations. Temporal means that the lifetime of a virtual organization is normally either explicitly or implicitly restricted; the virtual organization is dissolved when its overall goal has been achieved (see, e.g., [ByBP93]).

Camarinha and Afsarmanesh (see [CaAf99b], [CaAf99c]) identify different kinds and levels of virtual organizations by looking at it from five different angles/views:

- *Duration:* single business versus long term alliance; especially the first case requires an infrastructure that supports fast and dynamic consortium creations/dissolutions
- *Topology*: variable/dynamic structure versus fixed structure; in the first case member organizations may dynamically join or leave the alliance, e.g., in accordance to the phases of the business process or other market factors
- *Participation*: single alliance versus multiple alliances; in the multiple alliance case an organizations must be capable to deal with/obey the goals of all virtual organizations it is involved in
- *Coordination*: Star-like respectively centralized coordination structure versus democratic alliance versus federation; in the *star* case a dominant member organization exists which defines the rules of the virtual organization respectively coordinates the actions of it; in the *democratic* case all involved organizations cooperate on an equal basis, preserving their autonomy, but joining their core competencies; however, a coordinator may still be necessary, among others to monitor the organizational structure and joint operation principles; a federation is a democracy whose individual members believe that they can even improve their outcome by creating a kind of joint coordination structure.
- *Visibility*: single level versus multi-level; the visibility feature defines what a member organization is allowed to see from the structure of the virtual organization: only its direct neighbors (single level) or more (multi-level)

For our approach we do not make assumptions with respect to the above features. However, the following characteristics describe the virtual organization concept for which our approach works best:

- single business, variable structure, multiple alliances, democracy or federation and single level visibility.

The life cycle of a virtual organization is defined by

- the *creation/configuration phase*, in which the partners are selected and the possible relationships and contracts are negotiated and agreed on,
- the *operation phase*, in which the necessary information needs to be exchanged between and within the member organizations, unforeseen events and exceptions need to be handled with, and the operation processes need to be coordinated
- the *dissolution phase*, in which all necessary actions for the dissolution of the virtual organization are taken

In this paper we will mainly concentrate on the creation/formation phase. We especially will discuss how partners can be found and why and how the structure of a virtual organization may change constantly during the lifetime of a virtual organization.

2.2 Holonic Enterprises and Emergence

Since a virtual organization is only a temporary conglomerate that is established to quickly react to complex demands of the market it is mandatory that every decision on the inter-enterprise level (*horizontal* integration), is propagated through and reflected in all levels down to the lowest level, the machine level (*vertical* integration). This leads to the concept of a holonic enterprise.

About twenty-five years ago Arthur Koestler, a Hungarian philosopher, introduced the word *holon* to describe a basic unit of organization in social and biological systems (see [Koes67]). *Holon* is an artificial word, derived from the Greek word *holos,* meaning *whole,* and the suffix *on* meaning *particle* or *part*. Koestler observed that entirely self supporting, non-interacting entities do not exist as such in living organisms and in social organizations. Instead, every identifiable unit of organization, such as a single cell in an animal or a family unit in a society, comprises more basic units (plasma and nucleus, parents and siblings) while at the same time forming a part of a larger unit of organization (a muscle tissue or a community). A holon is an autonomous and cooperative building block of a system that has a unique identity, yet may be made up of sub-ordinate parts and in turn may be part of a larger whole. The goal of a holonic organization, called *holarchy,* is to attain the benefits that holonic organization provides to living organisms and societies. The concept of holons enables the construction of very complex systems that are nonetheless efficient in the use of resources, highly resilient to internal and external disturbances, and adaptable and flexible in the face of changes in the environment in which they exist. It combines the best features of hierarchical ("top down") and *heterarchical* ("bottom up", "cooperative") organizational structures as the situation dictates [JMS]. This concept can preserve the stability of hierarchy while providing the dynamic flexibility of *heterarchies*.
The stability of holons and holarchies stems from holons being self-reliant units, which have a degree of independence and handle circumstances and problems on their particular level of existence without needing to ask higher level holons for assistance. Holons can also receive instruction from and, to a certain extent, be controlled by higher level holons. This self-reliant characteristic ensures that holons are stable, able to survive disturbances. The subordination to higher level holons ensures the effective operation of the larger whole.
If we apply the holonic paradigm to enterprises we have to adapt the concept to the well-established layers of abstraction in the control of a flexible (holonic) manufacturing system:

- *Production planning and control level*
  This highest level within an enterprise is defined by all the holons that are present at a specific site of the company. On this level, among others, the planning process for a production process is initiated and controlled.
- *Shop floor level*
  This level contains all the holons for a specific production unit together with the agent that represents shop floor control.
- *Flexible cell level*
  This level comprises the holons that are formed by the physical systems that belong to the flexible cell and, additionally, the flexible cell control system.
- *Autonomous system/machine level*
  This layer is described by the physical body of an autonomous system (like a machine, a tool or an autonomous robot) together with its controlling unit/agent.

This hierarchy documents the principle architecture of a holonic enterprise. Each level is represented by one or a number of holons each of which representing a unit on this level. Such units recursively consist of sets of holons each of which being responsible for a subtask of the overall task of the holon at hand. These "sub-"holons do not need to be from lower levels only, but can as well be holons from the same level. For example, a flexible cell holon may consist of holons representing autonomous systems as well as flexible cells which perform tasks that are necessary for the flexible cell holon at hand in order for it to be effective.
Most of the holons on the different levels are pretty stable in their structure. An exception are the flexible cell holons. They represent a unit that is capable to perform a higher level task that consists of a number of subtasks that are to be performed in a cooperative and harmonized manner by a set of "lower order" flexible cells and autonomous systems/machines. Flexible cells need to be formed according to new and emerging needs or may reconfigure themselves in order to be more efficient or in case of a failure or unsatisfying performance of a member holon. So, we need efficient mechanisms to dynamically form new holons or reconfigure existing ones.

Emergence
On the basis of this notion of a *holonic enterprise* we want to develop and implement a model for highly efficient and competitive virtual organizations in cyberspace that, in addition to addressing the above requirements and features, exhibits *emergent* (*self-organizing* and *evolutionary*) behavior. The model will rely on patterns of self-replicating structures organized in nested hierarchies, namely the already introduced *holarchies*.

As result of the process of evolution driven by the law of synergy, *emergence* endows the dynamics of composite systems with properties unidentifiable in their individual parts. The phenomenon of emergence involves

- *self-organization* of the dynamical systems such that the synergetic effects can occur;
- interaction with other systems from which the synergetic properties can *evolve* in a new context.

Multi-agent systems enable cloning of real-life systems into autonomous software entities with a 'life' of their own in the dynamic information environment offered by today's cyberspace. The *Holonic Enterprise* (HE) has emerged as a business paradigm from the need for flexible open reconfigurable models able to emulate the market dynamics in the networked economy (see [MHWW95]), which necessitates that strategies and relationships evolve over time, changing with the dynamic business environment. Building on the MAS-Internet-Soft Computing triad to create a web-centric model endowing virtual communities/societies (generically coined as 'enterprises') with proactive self-organizing properties in an open environment connected via the dynamic Web, the holonic enterprise paradigm provides a framework for information and resource management in global virtual organizations ([Ulie02]). A *holonic enterprise* is a holarchy of collaborative enterprises, where each enterprise is regarded as a holon and is modeled by a software agent with *holonic* properties, so that the software agent may be composed of other agents that behave in a similar way but perform different functions at lower levels of resolution (see [UlBW02], [UlWB01]). According to this paradigm of holonic (virtual) enterprises, which - from our point of view - is the most sophisticated approach to virtual enterprises, several layers of collaboration/interaction can be identified.

Since we now deal with a conglomerate of enterprises we need to extend our holarchy discussed in the last section by one more level, the *Global Inter-Enterprise Collaborative Level* or inter-enterprise level for short. On this level, virtual enterprises form and evolve in order to produce a complex product or provide a sophisticated service. Each collaborative partner is modeled as an agent that encapsulates those abstractions relevant to the particular cooperation. By this, a dynamic virtual cluster emerges that – on the one hand – is supposed to satisfy the complex overall goal at hand and – on the other hand – considers the individual goals of each enterprise involved as good as possible. Such a dynamic collaborative conglomerate can cope with unexpected disturbances (e.g., replace a partner that cannot deliver within the deadline) through re-configuration of the open system it represents.
For reasons of simplicity the next layer, the *Intra-Enterprise Level*, is a subsumption of the already discussed production and planning, shop floor and flexible cell level. In order for an enterprise to be capable to take responsibility for the intended part of the overall goal, it has to find out about its own internal resources to ensure that it can deliver on time according to the coordination requirements of the collaborative cluster and with the required quality. Planning and dynamic scheduling of resources at this level enable functional reconfiguration and flexibility via (re)selecting functional units, (re)assigning their locations, and (re)defining their interconnections (e.g., rerouting around a broken machine, changing the functions of a multi-functional machine). This is achieved through a replication of the dynamic virtual clustering mechanism having now each resource within the enterprise cloned as an agent that abstracts those functional characteristics relevant to the specific task assigned by the collaborative conglomerate to the partner. Re-configuration of schedules to cope with new orders or unexpected disturbances (e.g. when a machine breaks) is enabled through re-clustering of the agents representing the actual resources of the enterprise (see Fig. 1). The main criteria for resource (re)allocation when (re)configuring the schedules are related to cost minimization achieved via multi-criteria optimization.
For an enterprise to be capable to contribute to a higher goal it first has to find out whether it can and wants to provide the necessary resources. For this reason functional units on the intra enterprise level are represented by agents that – in a collaborative effort – decide whether the enterprise can and should engage in a virtual enterprise.
The lowest level is the machine level. It is concerned with the distributed control of the physical machines that actually perform the work. To enable agile manufacturing through the deployment of self-reconfiguring, intelligent distributed automation elements (see Fig. 2) each machine is cloned as an agent that abstracts those parameters needed for the configuration of the virtual control system managing the distributed production.

How does one build agents and groups of agents which fulfill the holonic philosophy? If a one-to-one mapping of holon to agent is performed, it is much more difficult to practically implement an agent (then it is to conceptualize a holon) that itself may be a component of a higher level agent and/or may contain lower level agents. Here, the concept of a *mediator agent* comes into play. The mediator will fulfill two main functions. First, it acts as the interface between the agents in the holon and between the agents outside the holon (i.e. acts as a type of facilitator); conceptually, it can be thought of as the agent that represents the holon. Second, it may broker and/or supervise the interactions between the sub-holons of that holon; this also allows the system architect to implement (and later update) a variety of forms of interaction easily and effectively, thereby fulfilling the need for flexibility and re-configurability. The mediator encapsulates the mechanism that clusters the holons into collaborative groups (see [MaNo96]). The architectural structure in such holarchies follows the design principles for metamorphic architectures. With each collaborative partner modeled as an agent that encapsulates those abstractions relevant to the particular cooperation, a dynamic virtual cluster emerges which can be configured on-line according to the collaborative goals. Such a dynamic collaborative holarchy can cope with unexpected disturbances (e.g. replace a collaborative partner who breaks commitments) through on-line reconfiguration of the open system it represents. It provides on-line task distribution across the available partners as well as deployment mechanisms that ensure real-time error reporting and on-demand workflow/information tracking (e.g. fault tracking in distributed discrete manufacturing, etc.).

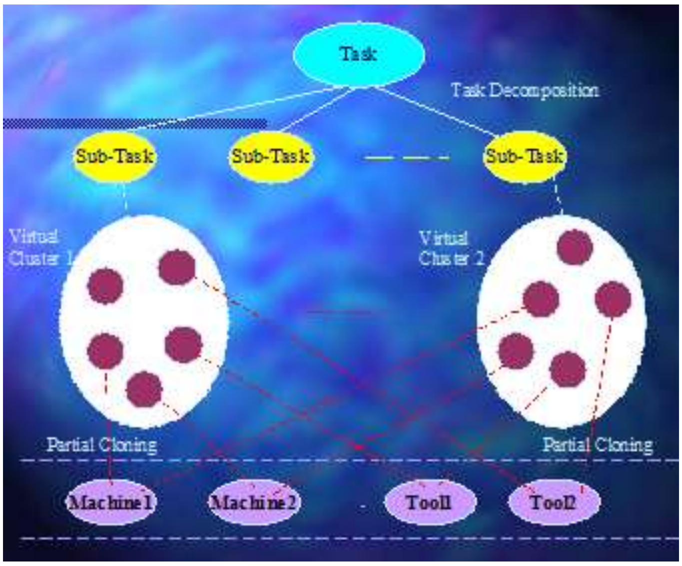


Fig. 1 *Task Distribution Pattern at the Intra-Enterprise level.*

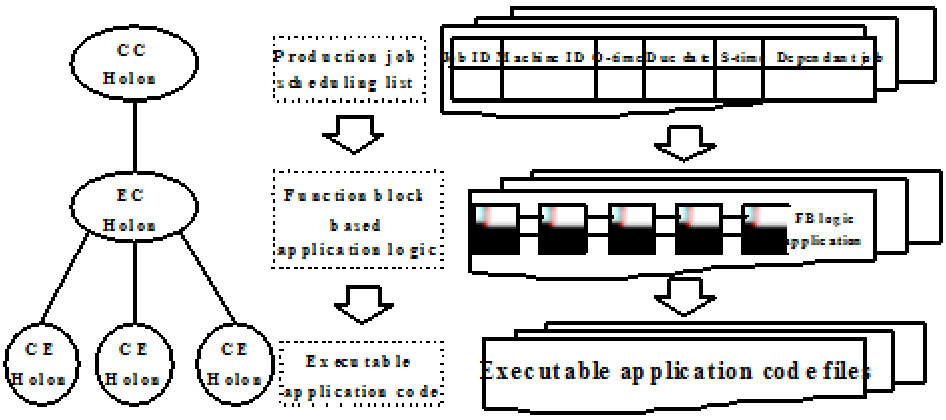


Fig. 2. *Task Deployment Pattern at the Virtual Control Level*

2.3 Patterns of Virtual Collaborations

The common mechanisms that characterize the collaborative information ecosystem created by the three levels of a virtual enterprise follow the design patterns for adaptive multi-agent systems identified by [ShSN99] (see Fig. 3).

Metamorphic Architecture Pattern. The overall architecture of the Virtual Enterprise builds on this pattern that replicates at all levels [UlWB01].
This pattern works by synergetic integration of two other patterns:

- *Dynamic Virtual Clustering* configured to minimize cost and enabling for flexible, re-configurable structures. At all levels of the virtual enterprise, task propagation occurs by a process of virtual cluster (or conglomerate) formation. This pattern is facilitated by the general layered architecture of the virtual enterprise. Each level previously described is divided into a number of autonomous layers that appear to interact through an application programming interface (API). Code is run asynchronously on these layers, providing functional separation between the layers.

- *Mediator Agent Pattern* supporting the decision-making process that creates and (re)-configures the collaborative cluster of enterprises.

To abstract those characteristics of the entities in each cluster that are relevant for the particular collaboration at each level we use the

- *Partial Cloning Pattern*. This pattern defines which of the enterprise's characteristics (attributes and functionality) we need to abstract into agents at each level when modeling the virtual enterprise as a collaborative multi-agent system.

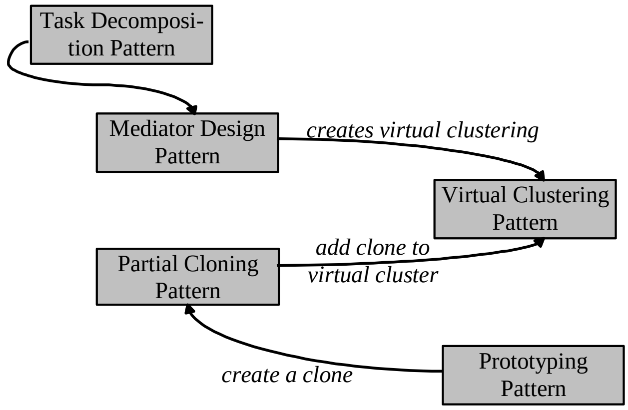


Fig. 3. Pattern Interaction within the Metamorphic Architecture

The workflow coordination through the collaborative ecosystem is managed by the mediator agent via the Task Decomposition-Distribution Pattern [MaNo96]. This pattern is enhanced with the capability to distribute harmoniously among the participants, the overall task assigned to the collaborative holon, at each level. The main mechanisms by which this pattern works are **task distribution** among the cluster's entities (outside-in view from the mediator "down" into each collaborative partner at that level) and task deployment within each entity (inside-out view – from the entity, regarded as a holon with distributed resources available to it for accomplishing the assigned task, to the mediator).
Propagation of the task decomposition-distribution pattern throughout the granular levels of the virtual enterprise requires two kind of ontologies to enable 'inter-entity' communication, which define an **Ontology Pattern**. This consists of two kind of ontologies, namely for 'peer-to-peer' communication at each level (that is 'inter-agent' communication among entities that form a cluster); and for 'inter-level' communication that enables deployment of tasks assigned at higher levels (by the mediator) on lower level clusters of resources.
The FIPA-Enabled Virtual Enterprise Project[1] merges the efforts of several international consortia (FIPA - www.fipa.org; HMS - http://hms.ifw.uni-hannover.de/; Agentcities – www.agentcities.com) to expand the existing FIPA architecture with new services supporting collaborative work in global virtual enterprises. In the sequel we will present the main aspects of the FIPA infrastructure of dynamic interactive interoperable services on which virtual services can be built.

3 Emergent Virtual Enterprise Implementation Using and Extending the FIPA Approach

A detailed description of the FIPA architecture is presented in [UlCN01]. Here we will focus only on the implementation details using the recent standards developed by the Foundation for Intelligent Physical Agents, in particular on the FIPA-OS platform (see [FIPA02]). The FIPA architecture provides an agent software framework and brings interoperability within and across agent based applications. It employs intelligent agents. The implementations are based on Java and Internet relevant technologies (XML, SMTP, Jini, Active Objects), elements which have already become regular "ingredients" in applications of agents. The interoperability

[1] The term Virtual Enterprise was coined by Dr. Mihaela Ulieru on April 3, 2001 at the FIPA meeting in London, UK where this focus was defined for the FIPA PDM Work Group.

involves relationships between agents, between agents and platforms and relationships between implementations of agent services.
The FIPA OS architecture, Fig. 4, proposes the concept of Agent Platform (AP) which is built on top of a distributed computing environment, thus integrating the client/server paradigm with agent technology.
As it results from Fig. 4, the AP consists of three basic services: the Agent Management System (AMS), the Directory Facilitator (DF) and the Agent Communication Channel (ACC). When agents are registered with the AMS for a platform they are considered being part of that platform. The AMS is responsible for the management of operations on the agent platform as well as for the management of the agents themselves. The DF (an agent itself) works in a yellow pages manner and supports the localization of agents and the services they provide in the area of a domain or the whole environment. In order for the agents to be registered with a DF, they have to send a registration request to that service. The ACC uses information provided by AMS to enable agent communication between agents on a platform and between platforms by offering a message forwarding service. The high level of interoperability between platforms is made possible by placing a mandatory ACC agent on each agent system belonging to a FIPA compliant environment.
To support IIOP (Internet Inter-Orb Protocol), which is the necessary protocol for interoperability, the ACC agent is enabled with a CORBA interface. Thus, as a CORBA object, the ACC agent is provided with methods that can be invoked via IIOP. The communication between FIPA agents takes place through the specific communication language, FIPA-ACL.

Different objects and/or programs can interact with one another through an ORB (Object Request Broker). Therefore, agents and applications will each have their own "brokers", which can carry on the message exchange through the IIOP specifications, for that scope using the TCP/IP layer that provides the port numbers and hostnames for reliable delivery. In other words, the ORB layer is responsible for directing and "translating" the IIOP requests.

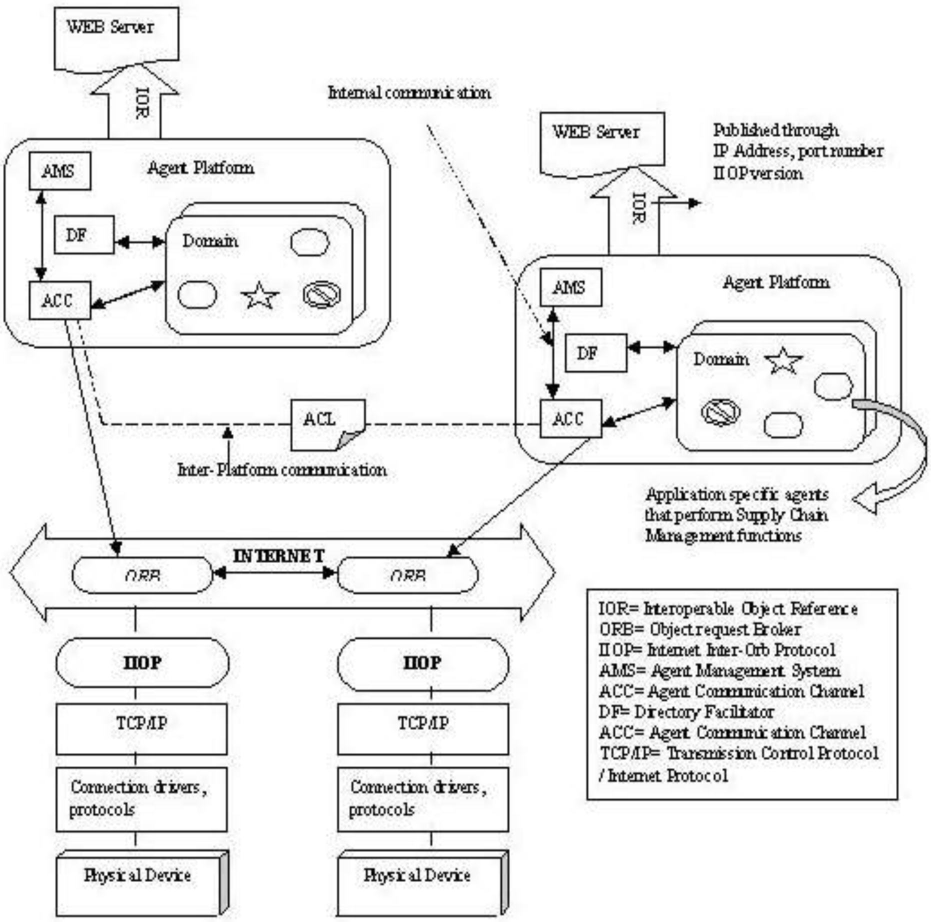


Fig. 4. The Main Elements of the FIPA-OS Architecture

With current implementations of ORBs, it is not possible to use the URL as an address for a platform; hence, the solution is to share a directory in which the platforms will have their own files containing the corresponding IORs.
The domain is a virtual space (area) that facilitates the management of different application entities (supplier, order, search, transport, etc.) and encapsulates certain capabilities and restrictions for residing agents providing uniformity in the interactions that take place between software modules (see Fig. 4).

In the sequel we will propose a way to extend the FIPA architecture with virtual services for global virtual organizations.

3.1 Forces to be balanced

Here the driving forces are triggered by the objectives of any relationship-based enterprise [UlWB01]:

Efficiency and cost minimization - achieved via maximum synergy (obtained by clustering the 'best[2]' partners). Efficiency is obtained by an openness to continuously sense the market's pulse and rapid (re)configuration to respond quickly to changes, as well as by the ability to respond to errors in a timely fashion. This in turn triggers new objectives:
On demand order tracking, on-line order error reporting, ability to quickly replace a collaborative partner if it does not fulfill its commitments in an appropriate way.

Competitiveness on the global market. The collaborative cluster can achieve competitiveness only through continuous optimization of the collaborative cluster with maximum synergy as criteria. If a partner doesn't

[2] Of course, finding the best partners is an NP-hard problem and can therefore not be achieved. Here and in the following when we use the term best we mean the best possible solution.

perform according to expectations (e.g. doesn't honor commitments, doesn't deliver on time, doesn't bid good enough to compete with its outside competitors) it will be replaced with a more suitable partner. This decision and appropriate negotiation will be performed by the mediator.
The driving forces are:

- Need for optimal clustering (i.e. always group the best partners) – requires on-line re-configuration of the collaborative cluster to respond to changes in market demands as well as to the needs for maintaining optimal configuration.
- Need to balance autonomy of each individual partner with the cooperative demands of the collaborative cluster – through negotiation that can range from simple bidding (proposal and counter-proposal) to complex argumentation and persuasion strategies. An example may be the following: the cluster sets a deadline and requirements to coordinate among the partners while partners need to argue their position and integrate the deadline with their other priorities. The cluster sets the 'rules of the game' through component protocols [VuJe00]. Preferences can be captured via a utility function such that clustering best partners can be achieved via cost minimization (e.g. via fuzzy entropy minimization [UlNo00]).

An extremely important issue related to inter-platform accessibility at this level is Security standards that would enable a fair balance of the autonomy and cooperative forces by enabling enough access to the collaborative cluster's entities to each-other services while keeping secrets safe. FIPA has a special Work Group dedicated to the investigation of security requirements for inter-enterprise business in dynamic service environments.

3.2 Patterns Particularities and Required Services

The need to balance these forces leads to the following pattern particularities that in turn demand for specific services:

Metamorphic Architecture pattern. A main requirement to implement the pattern at this level is to ensure inter-enterprise/inter-node/inter-platform communication among the participants in the collaboration. FIPA (www.fipa.org) has already developed strong services that enable each enterprise to become a Node in a Collaborative network of Agentcities (www.agentcities.org). Each enterprise is supposed to implement its software to run on FIPA compliant agent platforms such that agents on different platforms will be able to communicate with each other and access each other's services to create new value added services for the collaborative cluster.

Partial Cloning Pattern. The main attributes that each enterprise has to abstract into agents at this level are: provided goods and services with which it can enter the collaborative demand-supply game; marketing strategies [JFNO00] that is those related attributes and functions that enable company's integration into an existing cluster as well as it to be chosen when a new cluster is formed;

Mediator Agent Pattern. The decision-making particulars in this pattern are strongly determined by the abstractions made in the partial cloning pattern as well as by the implementation mechanisms of the task decomposition-distribution pattern. The main driver of the "inside-out" enterprise-to-cluster negotiation is obtaining the trust of the mediator in charge with the coordination of the collaborative cluster. In implementing a flexible utility function for the "outside-in" cluster-to-enterprise decisions, factors such as how much does the cluster need the services provided by the particular enterprise under evaluation; is the cost of keeping this partner worth keeping it or better replace. An interesting way to decide on selecting or keeping a partner is suggested by [FaSJ00]. They use fuzzy similarity to select the partner whose proposal is most similar to an opponent's last offer and whose trust degree is higher.

Virtual Clustering Pattern. To form and always keep a "best" cluster the mediator needs Grouping Policies (http://www-dse.doc.ic.ac.uk/Research/policies/) such as obligation, constraining and authorization that also enforce the security requirements on each partner [LSDD00] that enable nested management structures. Contractual frameworks that enable nested management structures [LuSl99] are essential clustering mechanisms that deal with autonomy in policy-restraining contexts and under security constraints. This is resonant with the concept of Cooperation Domain [UNKS00] introduced by Dr. James Christensen (http://www.holobloc.com/about.htm) in connection with the virtual control concepts developed by the Virtual Manufacturing Systems Consortium (http://hms.ifw.uni-hannover.de/).

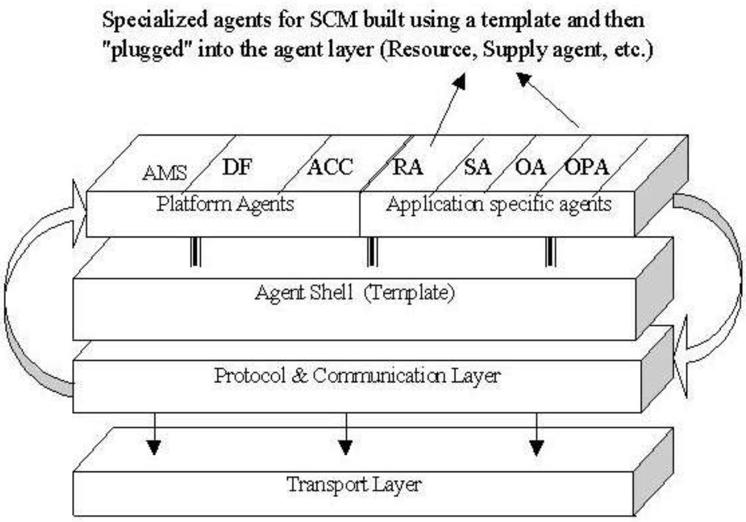
Task Distribution-Decomposition Pattern. At this level, of critical importance are the compliance mechanisms (such as "reputation" and "regimentation") that can be enforced by the mediators upon the partners to coerce them in fulfilling their obligations when they assume responsibility for the assigned task [MaNo96]. Complex normative concepts enable interactive contractual design based on control mechanisms such as influence as a negotiation framework that configures the collaborative cluster.

3.3 Mechanisms Needed to Implement the Required Services

On the basic Dynamic Systems Environment (DSE) services each enterprise can build with the FIPA-virtual standards to add the services and policies needed for virtual collaboration. Extended standards at the inter-enterprise level include:

- Specification of core competencies
- Process and workflow specifications
- Wireless access to information for e.g. on-line order tracking and error reporting on manager's cell-phone screen, either on demand or as proactive notification by the system.
- On-line banking and financial services among the partners in the cluster
- Coordination mechanisms such as order 'ready' reporting to synchronize with the work done by the other collaborative partners.

In the sequel we present specialized services that expand the FIPA standards to the global supply chain domain



(see Fig. 5).

Fig. 5. FIPA-Enabled Virtual Enterprise Architecture for Virtual Supply Chain Coordination

The combination of Multi-Agent technologies with the Web-centric paradigm opens wide perspectives towards the development of mechanisms for creation of virtual organizations with a flexible, self-organizing structure. Unfortunately, the current FIPA specifications do not currently address the implementation of web-Centric paradigm specific elements. However, FIPA-OS can be easily extended to cover the e-Business aspects by building agents to take over order taking, accounting and security tasks. Moreover, software agents can collect information on products and suppliers that could fulfill the orders, evaluate bidding, making decisions on them, negotiating terms of transactions, place orders and make payments to the parts involved in the e-Business network. To integrate all these technologies seamlessly there is a need to define universal standards for goods and services, consumer and vendor profiles, nature of services, secure payment mechanisms, and business electronic forms. A set of issues such as security, performance and failure may be handled by proper design, others like communication and interoperability between applications may be managed by so-called middleware solutions (CORBA, DCOM), additional software platforms (Voyager) or built-in language mechanisms (Java RMI, RPC) and Web specific meta-languages (XML).

3.4 Agents, Mediator Functionality and Web Services

One of the toughest problems with respect to the emergence of virtual organizations is to find/set up the right mixture of component organizations in order to maximize the overall outcome/impact. In principal several possibilities exist for the formation of a virtual organization. We assume that there exists one initiating agent which starts and is in charge of the formation process. This agent may either totally rely on its own knowledge and, therefore, directly contact possible partners or may use knowledge sources, like mediators or yellow pages, to find the right partners. In the first case the agent has to maintain its own knowledgebase of acquaintances. The advantage is that the agent is in full control of the whole process, which includes full control over the result (how well did the virtual organization work?). With the second alternative an agent looses part or full control about the formation process. Again there are two possibilities: either the agent transfers the problem and with it the control about the solution process completely to a mediator agent or it uses such agents in its problem solving process. This may mean that it either makes use of a yellow page service or only transmits sub-problems to a mediator agent. With the rapid propagation of web services this second group becomes much more attractive.
The mediator agent can be replaced by or/and make use of web services. A web service is any service that is available over the Internet, uses a standardized XML messaging system, and is not tied to any operating system or programming language (see, e.g., [Cera02]).

Web Services
Web Services are self contained, self described modular Internet-based applications, which can be published, found and used on the web and usually execute a specific business task. The infrastructural basis of the Web Services concept is formed by a small set of XML based standards. Web services are important cornerstones of emerging architectures since they enhance interoperability between heterogeneous information systems. Interoperability is needed for enterprise application integration (EAI) in order to connect separated systems quickly and at low costs. The current Web Service architecture is built around a small set of de facto standards for message transfer, Web Service description and Web Service discovery.
Since these business services are described and provided in a well-defined, universal way each of these self-contained services can easily be integrated with other services to create a (more) complex business process. In order to support this, web services provide some functionality that permits interested people/agents to define, describe, search for, and execute these specific services. This interoperability allows businesses to dynamically publish, discover, and aggregate a range of Web services through the Internet to more easily create innovative products, business processes and value chains. The web service architecture can be viewed in two ways. As a layered architecture it consists of the four layers 1. service transport, 2. XML messaging, 3. service description and 4. service discovery.

1. The lowest level, the *service transport*, is responsible for transporting messages between applications/agents/services. This layer provides protocols like HTTP, SMTP, and FTP.
2. The *XML messaging* layer encodes messages in a common format so that they can be understood at either end. Common technologies on this level are SOAP and XML-RPC.
3. The *service description* is responsible for describing the public interface to a specific web service. Currently, it is handled via the Web Service Description Language (WSDL).
4. The highest layer is the *service discovery*. While the description level provides a general description of a service the discovery layer offers concrete addresses for services as well as a concrete description of the corresponding interface. Moreover, it supports generic search and discovery mechanisms for web services. Today, service description is usually handled via the UDDI-protocol (Universal Description, Discovery, and Integration).

The second view on a Web service architecture is by examining the individual roles of each web service actor. Three roles can be identified, namely 1. the service provider role, 2. the service requester role, and 3. the service registry role.

1. The *service provider* implements a specific service and makes it available on the Internet.
2. The *service requestor* is any consumer of a Web service. It utilizes an existing web service by opening a network connection and sending an XML request.
3. The *service registry* is a logically centralized directory of services. It provides a central place where developers/agents can publish new services or find existing ones. It therefore serves as a centralized clearinghouse for companies and their services.

Agents can serve as service providers and/or service requestors. If a complex task is to be solved an agent may either directly contact one or more web service registries to find appropriate partners or contact a mediator agent which may have (meta) knowledge about service registries and the services they provide.
Since the access to web services is standardized the search for a specific service may be quite easy and, therefore, can directly be done by the agent that is searching for a service. However, the use of a mediator agent may still have advantages since the mediator agent may store information about the success/performance of a cluster of agents in performing a collaborative, complex task on the one hand and in the performance of specific service providers (agents) on the other hand. Since a mediator may gather this experience from a huge number of tasks it may get a much better overview about the performance of individual service providers/agents clusters than an individual agent. Therefore, it may be much better able to define and establish highly successful clusters of agents if a complex task is to be solved. Moreover, with the further development and success of web service flow languages, like WSFL from IBM, complex service bundles can already be defined, persistently stored, and constantly be optimized.
The Web Services Flow Language (WSFL, [Leym01]) is an XML-based language for the description of web services compositions. WSFL considers two types of such compositions:
The first type specifies the appropriate usage pattern of a collection of web services in such a way that the resulting composition describes how to achieve a particular business goal; typically, the result is a description of a business process.
The second type specifies the interaction pattern of a collection of web services; in this case, the result is a description of the overall partner interactions.
A disadvantage of the use of mediators by individual agents that are in charge of solving a complex task (for a customer) is

- the dependency on mediators that compromises in a way the autonomy goal of an agent and
- possible fees/rewards that are to be paid for the mediator service.

With the further refinement and standardization of web services agent technology and web services have the capability to form a perfect symbiosis. Agents can provide their services, can compose complex services from simpler ones, can deal with the high dynamicity of the Internet (services may appear/disappear) and can optimize existing clusters/workflows of highly complex services (emergence).

4 Objectives and Cooperative Problem Solving in Virtual Organizations

In this chapter we will concentrate on

1. what kind of goals need to be deeply embedded in an agent's belief system in order it to be capable to truly represent its underlying organization and how this can be achieved. This is discussed in the section about intentional problem solving.
2. how the process of a formation of a virtual organization can be organized.

In this chapter we will assume that each organization that is willing to participate in virtual organizations is represented by an appropriate agent and that, therefore, virtual organizations are formed by agents negotiating and agreeing on working cooperatively together to solve the complex problem at hand. A top-down approach will be presented which assumes that a virtual organization is formed on the initiative of one agent that tries to find other agents in order to be able to deliver the complex product/service bundle at hand. As discussed before, in principal such a formation process can either be supervised/controlled by a mediator agent or by the initiating agent itself. In both cases the agents may make use of yellow/white pages or similar services (like UDDI; as discussed in section 3.4) in order to find appropriate partners for cooperation.

Mediator service versus complete autonomy
Up to now we have always assumed that the formation process is done by a mediator agent. This has the advantage that only this agent needs to have all the abilities that are necessary to fulfill such a task, that is, the abilities

- to understand and interpret the complex problem at hand
- decompose it in sub-problems that can be dealt with by individual organizations
- organizing and dealing with the negotiation process with possible contractors/participants

- storing and maintaining a knowledge base in which experiences of the performance and efficiency of past virtual organizations and its members are stored

These abilities are pretty demanding and complex. Therefore, it will cost a lot of efforts and resources to implement them in each agent.

Agents representing representatives on the different layers of a virtual organization

In the next section we will concentrate on how a cluster of agents can be formed that acts in a collaborative way in solving a complex task. The complex task will be called the problem at hand and the process of dealing with it *multi agent problem solving*. We will abstract from the question whether a mediator is used and simply assume that the formation of a group is done by the problem solving agents themselves. In section 4.1 we will discuss the problem of planning while section 0 presents our approach of complex problem solving. Since we will discuss how agents may group in order to solve a complex problem/task that cannot be solved by an individual agent this process is more likely to occur on the global inter-enterprise collaborative level where a number of enterprises decide to work together for a limited time (form a virtual conglomerate) to solve the complex problem at hand. However, in large enterprises, where a number of departments may compete to solve tasks, this task of choosing/selecting the right agents for the complex problem solving process may occur as well.

Keeping in mind the agent's characteristic of autonomy, only a direct interconnection of all the agents that are to become members of a MAS is possible. Integration and indirect interconnection are incompatible with autonomy. This means that the underlying system must enable the agents to directly contact their acquaintances (peer to peer message) or to issue a non-directed broadcast to contact all agents that are known to the system.
Due to this requirement reasons can be given for the necessity of the acquaintance attribute each agent has to keep. Since the agents within such a MAS can not make use of a central "relay station" that knows all the agents and their capabilities so that received requests for cooperation can be routed to the best qualified agent, rather each agent itself must be aware of its colleagues and their respective qualities for possible cooperation. Therefore, an agent has to be able to memorize these agents it is able to get in direct contact with, together with a description of their characteristics what results in its personal "yellow pages".

4.1 Intentional Problem Solving

Within this section we will concentrate on a single organizational unit and its (problem solving) behavior. Such a unit can be a profit-center or an individual corporation. In order to provide complex products or service bundles first of all a comprehensive planning process is to be performed. To achieve flexibility and dynamicity the process needs to rely on the concept of modularized capabilities. Thus planning enables the identification of those elementary products or services that can be combined for product or service bundles. As is illustrated by Fig. 6 planning can be performed top-down, that is by problem decomposition[3], and bottom-up, that is by service or product aggregation.
To find the optimal solution for such a kind of planning in general is impossible since it is NP-hard. To nevertheless provide a solution simplifications or heuristics have to be applied which are supposed to be good enough to provide a solution that is at least better than the average. From the perspective of the organizational unit that alternative has to be chosen that promises the highest profit[4]. Therefore, the organizational unit has to open up its planning for this profit dimension in order to behave economically reasonable. Consequently, two different kinds of goals can be distinguished and described: *output* goal and *system* goal. According to [Perr70] the former stands for the plan goal that is to be achieved, that is the problem that has to be solved, and the latter describes the real purpose of the organizational unit, which often means profit maximization.

[3] It is important to understand that the decompositions have to completely cover the original problem. If, as a result of the decomposition process, a subsequent synthesis is required, it has to be encoded in one of the sub-problems.

[4] For reasons of simplicity we concentrate exclusively on profit. Of course, many other aspects can also be taken into consideration.

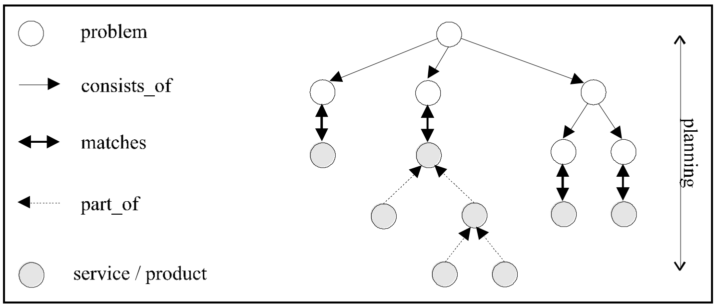


Fig. 6. Planning

An agent that represents such a unit must consider both goals. Systems showing a behavior that is dependent on certain goals in a reasonable way are characterized as intentional [Denn87]. Intentionality is a key characteristic of agent-based technology in DAI. In addition agent methodology genuinely addresses planning respectively deliberative behavior [WoJe95]. Therefore[5], we propose an agent-based approach for the realization of such an IS. Such an agent must consequently be capable of intentional problem solving. The plan that is derived for a problem at hand has to achieve both the output and the system goal.

The coexistence of two kinds of goals that have both to be ensured by the generated plan necessitates a two-staged planning process. Due to the overriding importance of the system goals in comparison to output goals the traditional planning process that mainly concentrates on the realization of the output goal has to be revised in a way that it reflects the system goals in an appropriate way. However, system goals, to become effective, must strongly influence the planning process as such. Planning processes can best be influenced if they are controlled by meta-planning processes, that is, processes on a more abstract level. In such an architecture the basic problem solving behavior of an agent can be illustrated as in Fig. 7 which visualizes the interdependencies between the most important entities involved.

An in-depth discussion of both planning and meta-planning procedures is beyond the scope of this paper, especially since traditional planning has already been intensively studied in literature. Instead, we will only concentrate on meta-planning. Meta-Planning has to ensure that, based on the system goals, a rational choice can be made among planning actions that constitute alternatives[6] with respect to the given output goal. Consequently meta-planning has to control the planning process either in a dynamic or in a static manner:

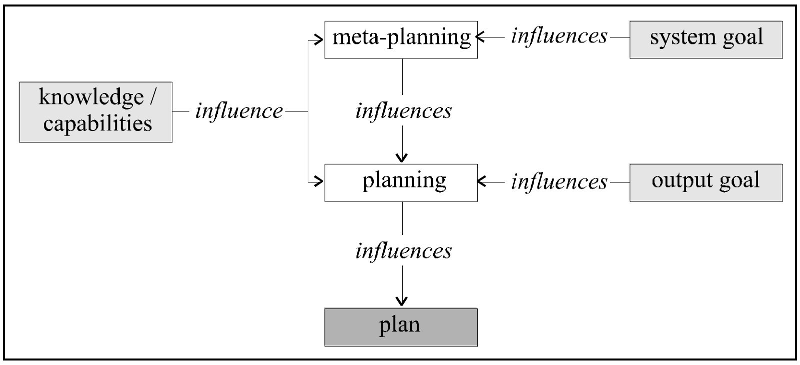


Fig. 7. Meta-Planning vs. Planning

Dynamic meta-planning stands for the direct control of the planning procedure. This means that whenever the planning process has to make a choice among alternative decompositions or, correspondingly, among alternative products or services the meta-planning process will determine the one from which it assumes that it is the most appropriate one with respect to the system goals.

5 As will be shown in the next section the intended inter-agent cooperation is another important reason for our agent-based approach since cooperation is an integral part of the MAS concept.

6 This choice among alternatives does not comprise all choices that are available to the planning procedure. Alternative in this context means that the respective products or services are likewise applicable within a given planning stage. Without meta-planning the planning procedure would use heuristics to decide on one of them, because from a planning perspective they equally serve the output goal. Otherwise meta-planning would actually comprise planning.

Static meta-planning describes the procedure where first of all a number of relevant alternative plans are generated. Subsequently that plan is chosen that reflects the system goals best. This corresponds to an extensive, however, non exhaustive[7] search through the set of all possible plans. This set represents the overall search space and is obviously relatively complex.
Both alternatives operate analogously on two subsequent levels of aggregation: Depending on the respective system goal both select one element from a set of alternatives. Dynamic meta-planning does so for a set of decompositions respectively services or products whereas its static counterpart works on a set of plans that again constitute a set of decompositions and products or services. Since the system goal has to enable a total ordering of the alternative elements it has to be defined by referencing some of their characteristics (e.g., profit contribution of each product). Consequently, system goals can be distinguished according to the kind of element they refer to. Thus dynamic meta-planning requires a system goal that is located at a decomposition / product or service level and static meta-planning necessitates system goals that are situated at plan level. Continuing analogously the aggregation by combining alternative plans leads to a system goal at a plan-set level.
Within the scenario of this paper such system goals can be formulated as follows:

Decomposition / product or service level:

i. "Make as much profit as possible by preferring certain products/services."

Plan level:

ii. "Make as much profit as possible from each task respectively problem."

Plan-set level:

iii. "Make as much profit as possible within a certain period of time."

Achieving a system goal at the plan-set level would constitute an "ex-post" meta-planning and can consequently not be implemented directly, because a runtime control of the planning procedure can only be realized by applying either dynamic or static meta-planning. However, both require system goals located at "lower" levels, i.e., plan or decomposition / product or service level. This motivates a top-down transformation of system goals that in general cannot be performed unambiguously[8] and consequently requires the use of heuristics[9].

ad i. Dynamic meta-planning can be applied, e.g. by prioritizing the decompositions respectively products or services so that the planning procedure obey the system goal as far as possible. This means it will apply the specific decomposition or use the specific product whenever possible for a given problem.

ad ii. Either static or dynamic meta-planning can be applied. Static meta-planning simply has to evaluate all plans that have been generated for the problem at hand and has to select the most profitable one. Dynamic meta-planning would have to rely on heuristics, since the system goal cannot be implemented directly at the product level. Such a heuristic could be to prefer always the most profitable product among a set of alternative ones.

ad iii. Either static or dynamic meta-planning can be applied but both have to rely on heuristics. Such a heuristic could be to prefer always the cheapest plan for the customer based on the assumption that most profit can be made if as much customers as possible can be attracted by extremely attractive offers. Static meta-planning would consequently select the cheapest plan out of the set of alternatives. Dynamic meta-planning would have to rely on a further heuristic, since the former cannot be implemented directly at the decomposition / product or service level. This second heuristic could analogously prefer always the cheapest product among a set of alternative ones.

Fig. 8 summarizes and visualizes the respective kinds of system goals.

---

[7] An exhaustive search would be NP-hard.

[8] In general it is not possible to derive the most profitable plan by simply choosing the most profitable decomposition or activity among emerging alternatives.

[9] Algorithm that can not guarantee an optimal performance in any case but an improvement for the average-case performance [Russell & Norvig 95, p. 94].

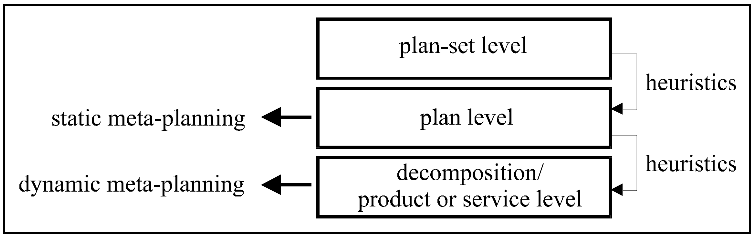


Fig. 8. System Goal Classification

4.2 Cooperation

By the combination of a number of agents for the creation of a MAS a prerequisite for cooperation has been made. A definition by Smith and Davis [SmDa81] for the cooperation of knowledge sources can analogously be applied to agents by simply substituting knowledge source by agent and information by the more general term of service. According to them agents cooperate in that sense that no one of them has sufficient capacity to solve the entire problem, mutual sharing of their services is necessary to allow the group as a whole to produce an answer.

4.3 Cooperative Problem Solving in MAS

An agent that is assigned a problem it may be able to solve will either completely or partly cooperate only for special reasons. Likewise the agents asked to cooperate will need some form of motivation, i.e. something serving their individual goals in order to engage in cooperation as autonomous agents [Cera02]. The reasons that will necessitate the initiation of a cooperation request are specified below as acting alternatives for the agent in charge of a problem. On the contrary the reasons that will cause an agent to accept a request are not elaborated on since acceptance depends exclusively on its private assessment with respect to its individual advantages.
The rough description of an agent's problem solving behavior is based on the assumption that agents are cooperating experts, i.e., have a certain area of expertise supplemented by the capability of recognizing their own responsibility for given problems. However, they are not specialists for only a certain class of problems but general problem solvers in their area of expertise. That means that in the absence of certain central problem specialists and with the aim of preventing a focusing to a certain class of problems each agent must be able to decompose problems itself. In doing so an agent is isolating that part of the problem it is able to understand from the rest. So each agent is able to decide whether and for what (part of a) given problem it is responsible. This capacity of decomposing problems is only restricted by the respective expertise of agents. It is not restricted to certain problems.

4.4 Multi-Agent Problem Solving (MAPS)

We will now expand our examination to the complete organizational compound and its problem solving behavior. The individual units are combined in order to enable an enlargement of the product or service spectrum by constructing more complex bundles. They are not integrated into a monolithic whole, but keep their (limited) autonomy in order to enable a flexible, i.e., demand-driven problem solving. Consequently, their cooperation cannot be based on a predefined structuring but evolves from negotiation. This process results in client (sub-)-contractor relationships for the time it takes to solve the specific instance of a problem.
Multi-agent problem solving enables a corresponding cooperation among the individual information systems that represent these organizational units. It realizes an inter-agent cooperation that is able to guarantee the autonomy of each component according to the decentralized character of the underlying organizational concepts, i.e. virtual organization or profit-center organization. The agents cooperate by coordinating only their output goals, they remain independent with respect to their system goals. The latter are exclusively determined by the respective unit[10] that is in charge of this subsystem and consequently can (will) differ from each other. According to Castelfranchi this is the important prerequisite for complete autonomy[11] [Cast95]. The flexible and demand-

[10] In case of a profit-center organization the system goal will correspond to the goal function according to which the performance of the profit center itself is assessed.

[11] Castelfranchi calls it "goal autonomy".

driven coordination of the output goals can be realized by supplementing the planning capabilities of each agent. As is illustrated by Fig. 6 an "isolated" agent that wants to solve an arbitrary problem has to find a matching product or service for each affiliated sub-problem. Now, within a multi-agent environment it is able to alternatively ask other agents for appropriate (sub-)solutions, if it is not able or willing to derive a solution itself. Therefore, an agent's planning results in a bipartite decomposition that additionally performs the separation of "internally solvable" and "open" sub-problems. The former are problems the agent itself is in charge of solving, while the latter are problems that are assigned to other agent (see Fig. 9):

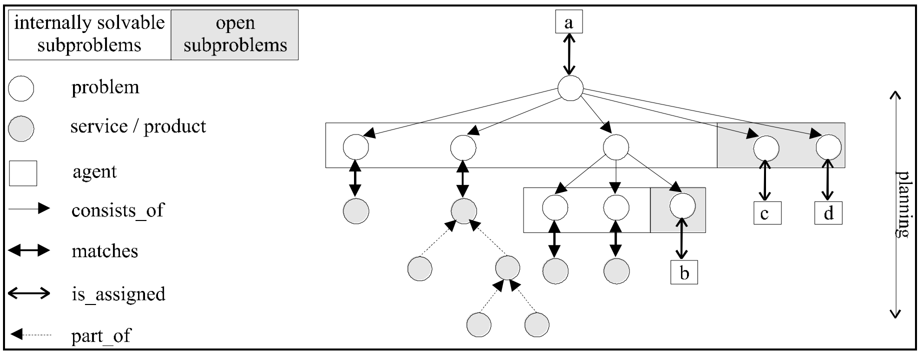


Fig. 9. MAPS Planning

This implies that an agent, in order to perform reasonable within a multi-agent context, is meant to be able to decompose also problems that are not part of its genuine problem solving expertise. Such a general purpose capability can only be realized by syntactic problem decomposition, i.e., decomposition that is exclusively defined by means of a problem description language and, in contrast to its semantic counterpart, does not require domain specific knowledge. An example is a state based domain model, where problems are defined via a pre- and a post-conditions and can consequently be decomposed by inserting any intermediate state[12]. Another one is the mode to represent financial problems and products by the stream of payments they define respectively realize. A decomposition is then performed by subtracting a "product stream" from a "problem stream".
The intended assignment of (sub-)problems to other agents must be the result of a bilateral negotiation process, since all participants are independent from each other. Each one will decide only on the basis of its individual system goals. Such a negotiation process can be realized by a contract net protocol, a negotiation scheme from DAI. The contract net protocol is supposed to realize a more flexible behavior of Distributed Problem Solving Systems (DPS) ([Smit80], [SmDa81]). It realizes a market like mechanism for the balance of interests by defining three subsequent phases respectively message types and implicitly three kinds of agent roles in the negotiation process (cf. Fig. 10).

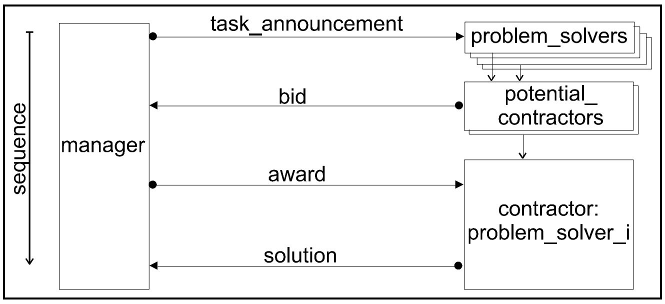


Fig. 10. Basic Contract Net Scheme

The fact that the components of a DPS system are considered neither intentional nor autonomous necessitates an adaptation of the above scheme in order to realize a true bilateral balance of interests. The Contract net assumes the (potential) contractor to be dependent on the manager and therefore a bid message provides no information about the potential contractor's demands on contracting. Consequently, the bid message has to be enlarged accordingly and the award message must be additionally interpreted as a manager's acceptance of these

[12] This permits no statements with respect to the solvability of the resulting sub-problems, since no domain-specific knowledge is used.

demands[13]. A further consequence that can be derived is that not the transfer of the solution will terminate the cooperation but a clearing phase that is in charge of realizing internal payments as the contractor's reward for providing the manager with the products or services.
Thus, MAPS is able to generate customized product or service bundles by flexibly coordinating the output goals of the participating agents. It establishes a demand-driven cooperation structure that consists of a number of client (≡ manager) contractor relationships and corresponds to the recursively descending problem decomposition that was applied. A more detailed discussion of MAPS can be found in [Wank97].

5 Conclusion

Assuming that information retrieval within organizations is an inherently distributed problem an information system concept was introduced that is based on the concept of MAS. A supplement was introduced and harmoniously integrated into the MAS concept to enable distributed problem solving. The results of this work are currently implemented in a system that is designed for giving advice to salespersons about product bundles a given virtual organization is able to offer.
Comparing the supplemented MAS concept with DPS with regard to the efficient solution of problems a severe drawback becomes apparent that motivates the contents of our future work:
Problem solution within DPS can start immediately and can be performed efficiently since organizational knowledge can be incorporated [von Martial 92, p. 17]. In contrast to that a MAS first has to derive such information (e.g. by Multi-agent Planning). Therefore, on the one hand DPS is more efficient in solving problems of a class it has been designed for than a MAS with comparable expertise. On the other hand the flexibility of DPS systems is severely restricted. Their applicability is limited to a single class of problems since organizational knowledge is partly wired up in their design (e.g. by communication paths between components).
The above statement motivates the question of how organizational knowledge can be memorized within MAS in such a way that their nature does not have to be sacrificed but can enable an improvement of their problem solving behavior[14]. The work we did in this field has led to two proposals:
The first solution relies on the concept of acquaintances of agents [Agha & Hewitt 88]. It enables each agent to store information about the characteristics of other agents from whom it is believed that they may be relevant for future cooperation. This would simulate agent-specific "yellow pages". They have to be accompanied by mechanisms that assure some degree of consistency with respect to its entries.
The second concept is that of explicit group representation through virtual agents [Wanka 96a]. A group that has implicitly emerged during the cooperative solution process of a given problem is made durable by representing it via a virtual agents. Such an agent is made out of references to the respective contributions that were made by each group member for the solution of that problem. Similar to the above proposal this concept must be likewise supplemented by mechanisms in charge of consistency (to avoid dangling references).

In the context of a global, web-driven economy, the capability of enterprises to flexibly join groups on the inter-enterprise level and form highly efficient groups on intra-enterprise is of high importance for the possible survival of enterprises. The paper presented the key concepts of a new e-Business model that merges latest results obtained by the Virtual Manufacturing Systems (HMS) Consortium with newly developed standards for platform interoperability released by the Foundation for Intelligent Physical Agents (FIPA), to enable the development of global collaborative e-Commerce/e-Business applications. We have shown that MAS technology and web services technology can perfectly supplement each other to provide a highly flexible and efficient environment in which even very specialized and demanding tasks can be solved. With Multi-Agent Problem Solving an approach has been presented which allows agents to cluster and collaborate in a way that fits best to the individual goals of each agent (e.g. maximum profit).
Currently we are working on a smoother integration of agent technology, web services and the specification of workflows on the Internet (web services flow). We believe that this will lead to highly competitive enterprises which can deal with all kinds of problems in a timely way.

[13] Of course, this does not yet enable negotiation cycles but it serves as a starting point.

[14] at least gradually through learning, i.e. the second time a similarly structured problem arrives

REFERENCES

[ByBP93] Byrne, J.; Brandt, R.; Port, O.: *The Virtual Corporation*. In: Business Week, 8. Feb. 1993, pp. 36-40.

[Cast95] Castelfranchi, Ch.: *Guarantees for Autonomy in Cognitive Agent Architecture*. In: Wooldridge, M. and Jennings, N. (eds.): Intelligent Agents. Lecture Notes in AI 890, Springer-Verlag, 1995, pp. 56 - 70.

[Cera02] Cerami, E.: *Web Services: Essentials*; O'Reilly Publishing Company; Feb. 2002

[DaMa92] Davidow, W.; Malone, M.: *The virtual corporation: structuring and revitalizing the corporation for the 21st century*. Harper-Collins Publishing Comp., New York, 1992.

[Denn87] Dennett, D. C.: *The Intentional Stance*. The MIT Press, 1987.

[FaSJ00] Faratin, P., Sierra, C., N. Jennings (2000) "*Using similarity criteria to make negotiation trade-offs*", Proc. 4th Int. Conf. On Multi-Agent Systems, Boston, pp. 119-126.

[FIPA02] FIPA-OS specifications and documentation at http://www.fipa.org/ & http:// www.nortelnet-works.com/fipa-os/

[MHWW95] McHugh, P., Wheeler, W., Merli, G. (1995). *Beyond Business Process Reengineering: Towards the Holonic Enterprise*, John Wiley & Sons, Inc., Toronto.

[Ioni97] Ionita, S.: *Genetic Algorithms for Control Engineering Applications* (in Romanian), ECIT-97, Pitesti, Romania, 21-22 Nov. 1997, p. 77-84.

[JFNO00] Jennings, N, R., Faratin, P., Norman, T.J., O'Brien, P. and Odgers, B. (2000) "*Autonomous Agents for Business Process Management*", Int. J. of Applied Artificial Intelligence 14(2), 145-190.

[Koes67] Koestler, A. (1967), The Ghost in the Machine, Hutchinson & Co, 1967

[Leym01] Leymann, Frank, *Web Services Flow Language*, (WSFL 1.0), IBM Software Group Report, May 2001, www-3.ibm.com/software/solutions/ webservices/pdf/WSFL.pdf

[LSDD00] E. Lupu, M Sloman, N. Dulay, N. Damianou: *Ponder: Realising Enterprise Viewpoint Concepts* 4th International Enterprise Distributed Object Computing Conference (EDOC2000), Makuhari, Japan, 25-28 Sept. 2000, pp.66-75.

[LuSl99] E. Lupu and M. Sloman "*Conflicts in Policy-based Distributed Systems Management*" IEEE Transactions on Software Engineering - Special Issue on Inconsistency Management, Vol 25, No.6 Nov. 1999, pp. 852-869.

[MaNo96] Maturana, F. and D.H. Norrie (1996) *Multi-agent Mediator Architecture for Distributed Manufacturing*. Journal of Intelligent Manufacturing, 7, pp.257-270

[Perr70] Perrow, Ch.: *Organizational analysis: A sociological view*. Wadsworth Pub Co; ASIN: 0818502878; (June 1970)

[ShSN99] Shu, Sudong and D.H. Norrie (1999) "*Patterns for Adaptive Multi-Agent Systems in Intelligent Manufacturing*", Proc. of the 2nd International Workshop on Intelligent Manufacturing Systems (IMS'99), Leuven, Belgium, pp. 67-74, September 22-24, 1999.

[SmDa81] R. Smith, R. Davis: *Frameworks for Cooperation in Distributed Problem Solving*, IEEE Transactions on Systems, Man and Cybernetics, Vol. SMC-11 1981, pp. 61-70.

[Smit80] Smith, R.: *The contract net protocol: High-level communication and control in a distributed problem solver*, IEEE Transactions on Computing, Vol. C-29 1980, pp. 1104 – 1113

[UlCN01] Ulieru, M, Cobzaru, M. and Norrie, D. *A FIPA-OS Based Multi-Agent Architecture for Global Supply-Chain Applications*, IPMM 2001 International Conference on Intelligent Processing and Manufacturing of Materials, July 29-August 3, 2001, Vancouver, BC.

[UlNo00] Ulieru, M. and D.H. Norrie (2000), "*Fault Recovery in Distributed Manufacturing Systems by Emergent Virtual Re-Configuration: A Fuzzy Multi-Agent Modeling Approach*", Information Science, 7669, ISSN # 0020-0255, September 2000.

[UlWB01] Mihaela Ulieru, Scott Walker and Robert Brennan, "*Virtual Enterprise as a Collaborative Information Ecosystem*", Proc. Workshop: Holons, Autonomous and Cooperative Agents for the Industry, Montreal, Canada, May 20, 2001 (AA 2001).

[UNKS00] Mihaela Ulieru, Douglas Norrie, Rob Kremer and Weiming Shen: *A Multi-Resolution Collaborative Architecture for web-Centric Global Manufacturing*, Information Science, Volume 127, Journal no. : 7669, ISSN # 0020-0255, August 2000.

[Ulie02] Ulieru, M. (2002). "*Emergence of Holonic Enterprises from Multi-Agent Systems: A Fuzzy-Evolutionary Approach*", Invited Chapter in *Soft Computing Agents*, (V. Loia; Ed.), IOS Press , 2002 (in print).

[UlBW02] Mihaela Ulieru, Robert Brennan and Scott Walker, "*The Holonic Enterprise – A Model for Internet-Enabled Global Supply Chain and Workflow Management*", International Journal of Integrated Manufacturing Systems, No 13/8, 2002, ISSN 0957-6061.

[VuJe00] Vulkan, N. and N.R. Jennings (2000), "*Efficient mechanisms for the supply of services in multi-agent environments*", Int. J. of Decision Support Systems 28(1-2), pp. 5-19.

[Wank97] Wanka, U.: *Multiagent Problem Solving*; PhD-Thesis; University of Essen; 1997

[WoJe95] Wooldridge, M.; Jennings, N.: *Agent Theories, Architectures, and Languages: A Survey*. In: Wooldridge, M. and Jennings, N. (eds.): Intelligent Agents. Lecture Notes in AI 890, Springer-Verlag, 1995, pp. 1 – 39

[CaAf99a] Camarinha, L.; Afsarmanesh, H.: *Infrastructures for Virtual Enterprises: Networking Industrial Enterprises*; IFIP 27; Kluwer Academic Publishers; ISBN: 0-7923-8639-6; 1999

[CaAf99b] Camarinha, L.; Afsarmanesh, H.: *The Virtual Enterprise Concept*; in [CaAf99]; 1999

[CaAf99c] Camarinha, L.; Afsarmanesh, H.: *Tendencies and General Requirements for Virtual Enterprises*; in [CaAf99]; 1999

[CaAf01] Camarinha, L.; Afsarmanesh, H.: in: Multi-Agent Systems and Applications, 9th ECCAI Advanced Course; ACAI 2001 and Agent Link's 3rd European Agent Systems Summer School, EASSS 2001, Prague, Czech Republic, July 2-13, 2001, Selected Tutorial PapersLNAI 2086, p. 335 ff.

[LiYe01] Liu, J.; Ye, Y (Eds.): E-*Commerce Agents: Marketplace Solutions, Security Issues, and Supply and Demand*; Springer Publishing Company, Lecture Notes in Artificial Intelligence LNAI 2033; 2001

[TiUn02] Tianfield, H.; Unland, R. (guest eds.) Special Issue "*Virtual Organisations and E-Commerce*"; International Journal of Information Technology and Decision Making; World Scientific Publishing Co.; Vol. 1, Issue 3, Sept. 2002

[TiBU02] Tianfield, H.; Binder, Z.; Unland, R.: *Multi-Agent Interactions under Organizational Hierarchy*. 2nd IFAC/IFIP/IEEE Conference on Management and Control of Production and Logistics (MCPL'00), Grenoble, France, July 5-8, 2000

[TiUn99a]Tianfield, H.; Unland, R.: *On the Hierarchical Structure of Multi-Agent Systems;* 1st International IFAC/IFIP/IFORS Workshop on Multi-Agent Systems in Production (MAS'99), Vienna, Austria, December 2-4, 1999

[TiUn99b] Tianfield, H.; Unland, R.: Formulating Enterprise Federation Into Multi-Agent Systems; 1st International IFAC/IFIP/ IFORS Workshop on Multi-Agent Systems in Production (MAS'99), Vienna, Austria, December 2-4, 1999

[ABHL02] DAML-S Coalition: Anupriya Ankolekar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David Martin, Drew McDermott, Sheila A. McIlraith, Srini Narayanan, Massimo Paolucci, Terry Payne, Katia Sycara: *DAML-S: Web Service Description for the Semantic Web*; in [HoHe02]; 2002

[Peer02] Joachim Peer: *Bringing Together Semantic Web and Web Services*; in [HoHe02]; 2002

[HoHe02] Ian Horrocks; James Hendler (Editors): *Proc. First International Semantic Web Conference (ISWC 2002);* Sardinia, Italy; 9-12th June, 2002, Springer Lecture Notes in Computer Science (LNCS) 2342; 2002

[LMST02] Michael Luck; Vladimír Mařík; Olga Stěpánková; Robert Trappl (Editors): *Proc. Advanced Course on Artificial Intelligence (ACAI 2001) "Multi-Agent Systems and Their Applications (MASA)",* Prague, Czech Republic; July 2001; Joint event of ECCAI (European Coordinating Committee for Artificial Intelligence) and AgentLink (European Network of Excellence for Agent-Based Computing); Springer Lecture Notes in Artificial Intelligence (LNAI) 2086; 2002

[FiFR02] Klaus Fischer, Petra Funk, and Christian Ruß; *Specialized Agent Applications*; in [LMST02]; 2002

[Lase98] T.M. Laseter. *Balanced Sourcing: Cooperation and Competition in Supplier Relationships*. Jossey-Bass, October 1998. ISBN: 0787944432.L