The ANEMONA + JANUS approach for engineering Manufacturing Enterprise IS

Adriana Giret

Departamento de Sistemas Informaticos y Computacion,
Universidad Politecnica de Valencia
agiret@dsic.upv.es,
WWW home page: http://www.dsic.upv.es/~agiret

Abstract. Open Manufacturing approaches are becoming more and more the appropriate tool and technology to meet the high expectations of the customers of manufacturing enterprises, who, in today’s economy, demand absolutely the best service, price, delivery time and product quality. Nowadays, Internet technologies have changed the way companies have conducted businesses and IS to operate their enterprise as well as manage customer relationship. Collaborative manufacturing environments demonstrate considerable potential in responding to this need. In this work we present a Multi-agent System approach, which integrates specific features of Intelligent Manufacturing Systems with Service Oriented Multi-agent technology, and a robust and complete software development method, for engineering Manufacturing Enterprise IS.

Keywords: Software Engineering Method, Intelligent Manufacturing Systems.

1 Introduction

In a relative short period of time, Internet technologies, mainly Service Oriented Computing, have touched upon and changed the way companies have conducted businesses to utilize Internet technologies to operate their enterprise as well as manage customer relationship. The convergence of Internet and manufacturing systems provides the basis for the creation of a new generation of computing solutions that can dramatically improve the responsiveness of organizations to better communicate with their customer and suppliers. Moreover, this new situation makes possible the rapid and easy on-demand creation of virtual manufacturing enterprises (open manufacturing systems) made up of different manufacturing partners that collaborate in order to fulfill the customer needs. Collaborative manufacturing environments demonstrate considerable potential in responding to this need. A collaborative environment integrating diverse information systems can enable the creation of ”virtual” enterprises with competencies to effectively and efficiently share their knowledge and collaborate with each other in order to compete in a global market.

In order to be successful in the world of e-Business and the discipline of e-Manufacturing, companies must still align their business initiatives with their
Adriana Giret

manufacturing flexibility. To make e-Manufacturing and supply chain technologies effective, integration is needed between various, often disparate technologies and manufacturing systems. The environment should automate individual manufacturing efforts, and mediate between individual companies to promote collaboration within the context of a manufacturing process. Many manufacturing paradigms promise to meet the challenges of the new manufacturing. Two of these paradigms, namely, distributed intelligent manufacturing systems and Holonic Manufacturing Systems [1], have recently been receiving a lot of attention in academia and industry.

Multi-agent system (MAS) represents one of the most promising technological paradigms for the development of open, distributed, cooperative, and intelligent software systems. Moreover, the areas of Service Oriented Computing and Multi-agent Systems are getting closer and closer. Both trying to deal with the same kind of environments formed by loose-coupled, flexible, persistent and distributed tasks [2]. An example of this fact is the new approach of Service Oriented Multi-agent Systems (SOMAS). Over the last twenty years, researchers have been applying agent technology to areas such as manufacturing enterprise integration and supply chain management, manufacturing planning, scheduling and execution control, materials handling and inventory management, to name a few. For an extensive literature review of these applications see [3].

An agent-supported manufacturing system is a manufacturing system consisting of autonomous modules (agents/holons [4]) with distributed control for transforming, transporting, storing and/or validating information and physical objects. The system integrates the entire range of manufacturing activities from order booking through design, production, and marketing to realize the agile manufacturing enterprise.

An e-Manufacturing environment or virtual enterprise system is a large-scale and complex system which development process has to be guided by specific, robust and complete software engineering methods. In this paper we present the ANEMONA + JANUS approach to develop this kind of systems. It is a complete software development process, which assists the system developer in the entire life cycle of the e-Manufacturing system, from requirement analysis to implementation. The approach is specific for Intelligent Manufacturing Systems in which the virtual enterprise is seen as a collaboration scenario populated by autonomous entities that interact by means of Web services.

2 ANEMONA

ANEMONA [5] is a MAS methodology for Holonic Manufacturing System (HMS) analysis and design that is based on the Abstract Agent notion [6] and the HMS modeling requirements [1]. ANEMONA integrates features from HMS, MAS and Enterprise Modelling techniques [7, 8] to engineer virtual enterprises as e-Manufacturing environments. In our approach an e-Manufacturing system is a HMS made up of service-oriented agents.
Adriana Giret

In ANEMONA, the manufacturing system is specified by dividing it into more specific characteristics that form different views of the system. These views are defined in terms of MAS technology; therefore, we talk about agents, roles, goals, beliefs, organizations, etc. We use Abstract Agent and holon as similar notions [4]. The views [9, 10] can be considered as general MAS models that can also be applied to other domains. The agent model is concerned with the functionality of each Abstract Agent: responsibilities and capabilities (services). The organization model describes how system components (Abstract Agents, roles, resources, and applications) are grouped together. The interaction model addresses the exchange of information or requests between Abstract Agents, collaboration scenarios. The environment model defines the non-autonomous entities with which the Abstract Agents interacts. The task/goal model describes relationships among goals and tasks, goal structures, and task structures.

In Figure 1 we can see the development stages of ANEMONA. The first stage, System Requirement Analysis and the second stage Holon Identification and Specification define the analysis phase of our approach. The aim of the analysis phase is to provide high-level HMS specifications from the problem Requirements, which are specified with the help of the problem domain experts and which can be updated by any development stage. The analysis adopts a top-down recursive approach. One advantage of a recursive analysis is that its results, i.e the Analysis Models, provide a set of elementary elements and assembling rules. ANEMONA provides specific guidelines [5] to help the engineer during the analysis phase. The HMS UC Guidelines can be used to identify and specify Use Cases, while the PROSA Guidelines are used to identify and specify PROSA types of agents [12]. The next stage in the development process is the Holon Design stage which is a bottom-up process to produce the System Architecture from the Analysis Models of the previous stage. In this phase, the engineer can use three guidelines [5], depending on the type of the MAS system to control the factory. The SIMBA Guidelines are for the SIMBA MAS implementation platform [13]. The JADE Guidelines are for the JADE agent platform [14], and the Function Block Guidelines are for implementing the physical processing part [15] of the factory resource agents. The aim of the Holon Implementation stage is to produce an Executable Code for the SetUp and Configuration stage. Finally, maintenance functions are executed in the Operation and Maintenance stage.

3 JANUS

The JANUS platform [16] is specifically designed to deal with MAS and HMAS (Holonic Multi-agent System). The metamodel of this platform corresponds to a fragment of the CRIO metamodel [17]. JANUS is a FIPA [18] compliant agent platform, designed to facilitate the transition between design and implementation phases. It thus provides direct implementation of: organization, group, role, holon and capacity. The organization is implemented as a first-class entity, which

---

1 The specification of the development process of ANEMONA is presented using SPEM diagrams [11].
Adriana Giret

Fig. 1. Stages of the methodology

includes a set of role classes. A role is local to a group (an organization can be instantiated in the form of groups), and provides holons playing the role the means to communicate with other group members. An agent is represented by an atomic holon (a non-composed one). JANUS defines two main types of holon: HeavyHolon and LightHolon. A HeavyHolon has its own execution resource (one thread per holon), and can therefore operate independently. The LightHolon is associated with synchronous execution mechanisms and it is very useful to develop multiagent-based simulations. A holon can play simultaneously multiple roles in several groups. It can dynamically access to new roles and leave ones that are no longer in use. To access or leave a role, a holon must meet the access and liberation conditions of the role.

The notion of capacity enables the representation of holon competences. Each holon has, since its creation, a set of basic skills, including the ability to play roles (and therefore communicate), to obtain information on existing organizations and groups within the platform, create other holons, and obtain new capacities. The capacity concept is an interface between the holon and the roles it plays. The role requires some capacities to define its behaviour, which can then be invoked in one of the tasks that make up the behaviour of the role. The set of capacities required by the role are specified by the role access conditions. This concept is the operational representation of the concept of service defined in the Service Oriented MAS.

The architecture of the JANUS platform is shown in Figure 2. The heart of the platform is embodied by its kernel, which provides the implementation of the organizational model and of the concept of holon. The different features provided by the JANUS kernel are used for: organizational management and the organization instantiations in the form of groups; the holon management system, includes all the tools required for the management of the holon life cycle; the communication channel controls the exchange of messages within the platform; the identification management system manages the neces-
sary mechanisms for assigning unique addresses; the directories/repositories maintains a directory of all the groups, organizations and holons defined in the kernel; the holon scheduling and observation management system provides the basic policies for holons scheduling; the logging system facilitates debugging processes.

4 The ANEMONA + JANUS approach

The ANEMONA implementation phase includes a set of guidelines and activities to deploy the manufacturing system into any FIPA compliant agent platform. Nevertheless, traditional FIPA compliant agent platforms do not provide built-in support for virtual organization management, and holonic structures execution. In this way, the management of these kind of structures need to be implemented from scratch by the developer, and controlled by dedicated entities that have to be specially designed for this purpose. JANUS overcomes this situation by including a set of primitive functions for holonic management and execution support. To take advantages of the JANUS features when developing e-Manufacturing systems with ANEMONA, it is required to connect these two approaches. In this way the set of management activities for the execution of the holonic structures of any e-Manufacturing can be delegated to the agent platform.

In order to define the ANEMONA + JANUS approach for e-Manufacturing systems it is required to: adapt the ANEMONA meta-models and the JANUS underlying meta-model; the JANUS execution entities have to be related with the ANEMONA analysis and design entities; and, a set of ANEMONA specific JANUS Guidelines have to be defined in order to help the system engineer when deploying the HMS into the JANUS platform.

The unified ANEMONA + JANUS meta-model is defined taking into account the central entities of both approaches. In ANEMONA the central entities are:
Abstract Agent, Role, Task, Organization, Agent, Goal, Interaction, and the relations among these central entities. On the other hand, the JANUS central entities are: Abstract Holon, Heavy Holon, Light Holon, Abstract Role, Capacity, Organization, Group, and Interaction. Figure 3 shows the unified ANEMONA + JANUS meta-model. In this figure, we can see that we have adapted/merged some entities of JANUS into the corresponding entity of ANEMONA (light grey boxes), and we have adopted/extended some entities of JANUS as new entities in the ANEMONA meta-models (dark grey boxes).

The Organization entity models a group of Abstract Agents (A-Agent in Figure 3), which cooperate to achieve common goals. This is an ANEMONA and JANUS entity, so its definition is common for the two approaches. An Abstract Goal (A-Goal) can be a Group Goal or an atomic Goal. A Group Goal is used to represent the goal of a Group entity. The Group entity is adopted from JANUS, it represents the implementation of an organization and contains a set of instances of different Roles. The Role entity is an ANEMONA entity that unifies the JANUS entities: Abstract Parallel Role and Holonic Role. The Abstract Agent is an ANEMONA entity that merges the Abstract Holon entity of JANUS. An Abstract Agent is an autonomous entity, it represents a non-atomic holon that is in turn composed of holons. The Heavy Abstract Agent and the Light Abstract Agent are JANUS entities to define types of holons. A Heavy Abstract Agent has its own execution resource (one thread per holon), and can therefore operate independently. The Light Abstract Agent is associated with synchronous execution mechanisms and it is useful to develop multiagent-based simulations.

Finally the Abstract Task entity integrates the notions of Tasks of ANEMONA, Capacity of JANUS and Services of Service-Oriented Architectures. The notion of Tasks/Capacity/Services enables the representation of holon competences. That is a set of basic skills, including the ability to interact by means of Web services. These entities are the interface between the holon/agent and the roles it plays. The role requires some Capacities/Services to define its behaviour, which can then be invoked in one of the Tasks that make up the behaviour of the role. In this way, the unified ANEMONA + JANUS meta-model integrates the enti-
ties of the two approaches in order to build on top of it a unified development approach compliant with the JANUS execution platform.

Figure 4 shows the correspondence among the ANEMONA analysis entities and the JANUS execution entities. This figure presents a sample IS for controlling a given manufacturing system. The diagrams with the analysis entities and its corresponding execution entities in the JANUS platform make clear the simple and direct correspondence among the modeling entities and the execution entities. In this way the Implementation phase of any manufacturing system is as simple as implementing all the Abstract Agents identified in the analysis and design phases of the ANEMONA + JANUS approach into Abstract Holons of the JANUS platform. The management of the different abstraction levels (see the levels in Figure 4) is delegated to the platform, in this way the developer should simply use the facilities that JANUS provides for the management activities, instead of implementing special controlling entities to monitor the correct execution of these levels.

In order to provide complete guidelines to the engineer when using the ANEMONA + JANUS approach it is required to define a set of specific JANUS Guidelines. These guidelines will help the engineer during the Holon Design phase of the development process (see Figure 1). The JANUS Guidelines will guide the engineer to produce the JANUS holon template. Figure 5 depicts the JANUS holon template. On the other hand, Table 1 shows the JANUS Guidelines. These guidelines connect the analysis and design entities of ANEMONA with the JANUS execution entities.

The Design Phase of the ANEMONA + JANUS approach includes the activity: Specify JANUS holon template. In this activity the software engineer has to complete a JANUS holon template for each agent in the Design Models.
Adriana Giret

Fig. 5. The JANUS holon template

this way, in the Implementation phase the engineer should only translate the set of JANUS holon templates into Abstract Holons execution code of the target JANUS platform.

5 Case study

The ANEMONA + JANUS approach was successfully used to implement a prototype of an e-Manufacturing environment for Open Manufacturing Systems called ASeM (Agent Supported e-Manufacturing). In ASeM we look into open agent-based technology to support Web-based collaboration between organizations within open virtual enterprises.

An ideal ASEm scenario for open virtual enterprises works as follows. Whenever a possible manufacturing partner (from any manufacturing enterprise) wants to sell and/or purchase a manufacturing capability (in our approach, a manufacturing capability is represented as a service) it initiates a collaboration scenario in the e-Manufacturing environment. The environment supports and allows the collaboration process all the way the entire life cycle of the given open manufacturing process. In such a process, when a collaboration initiative is initialized the e-Manufacturing environment requires the initiator enterprise to register the request to create a new collaboration scenario. The registration requires: the ID of the virtual partner (the manufacturing company), and the advertising service specification. The service specification includes the service ID, service profile as a WSDL document (that describes the service in terms of its inputs, outputs, preconditions and effects), service process (describing how a client has to use it), service grounding (supported message and binding details), a public private modifier to restrict access to the service (e.g. only a particular group of manu-
Table 1. JANUS Guidelines

<table>
<thead>
<tr>
<th>ID</th>
<th>Guideline</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>For the information processing part of every agent the JANUS Holon Template of Fig. 5 has to be completed following guidelines 2 to 9.</td>
</tr>
<tr>
<td>2</td>
<td>Item 1 of the JANUS Holon Template has to be filled in with a reference to the agent name in the Design Models of ANEMONA.</td>
</tr>
<tr>
<td>3</td>
<td>Item 2 of the JANUS Holon Template has to be filled in with the platform name to which the agent pertain.</td>
</tr>
<tr>
<td>4</td>
<td>In order to complete item 3 of the JANUS Holon Template the Agent Model, the Tasks and Goals Model and the Interaction Model of ANEMONA have to be used. Items 3.1 and 3.2 must be completed with all the Tasks a given agent offers as Services. Item 3.2 must indicate whether the service/capacity is offered for internal holons only, or it is also available for holons of other platforms. To do this the engineer has to analyze the Interaction Model of ANEMONA, particularly those interactions in which the agent participates.</td>
</tr>
<tr>
<td>5</td>
<td>In order to complete item 4 of the JANUS Holon Template the Agent Model and the Organization Model of ANEMONA have to be used. Items 4.1, 4.2 and 4.3 must be completed with all the Roles the given agent plays. Item 4.3 must indicate the set of capacities of the roles.</td>
</tr>
<tr>
<td>6</td>
<td>In item 5 of the JANUS Holon Template, the Software Engineer has to specify the elements that the holon can use in the message content (ontology). To do this, the engineer has to analyze the Agent Model and the Interaction Model in order to register the informational mental entities and the content of the messages in the interactions. Item 5.1 must be filled in with the concept name, 5.2 with the name (or names) of one (or more) base ontology, and item 5.3 with the schema which defines the concept structure.</td>
</tr>
<tr>
<td>7</td>
<td>In item 6 of the JANUS Holon Template, the Software Engineer has to specify the set of messages that the holon can interchange. To do this the engineer has to analyze the Interaction Model, mainly the interaction units that the holon may interchange. Item 6.1 must be filled in with the message name, 6.2 with the interaction unit, 6.3 with a reference to the interaction name, and 6.4 with the participation type (initiator, collaborator).</td>
</tr>
<tr>
<td>8</td>
<td>In item 7 of the JANUS Holon Template, all the Groups the holon participates in have to be completed. To do this the engineer has to analyze the Organization Model of ANEMONA.</td>
</tr>
<tr>
<td>9</td>
<td>Finally, item 8 of the JANUS Holon Template lists all the Groups the given holon controls. The Organization Model and Agent Model resumes this information.</td>
</tr>
</tbody>
</table>

Manufacturing partners can access a private service, Quality of Service (QoS) data, and the ontology used. This service specification is based on OWL-S specification for semantic Web services\(^2\). In this way the service becomes advertised, and whatever virtual partner, wants to collaborate with, may enter the collaboration scenario and complete the value chain in order to get an individual or global outcome in doing it. In order to complete the value chain of the collaboration scenario multiple cooperation environments can be initiated to fulfill the entire scenario goal. A holonic agent-based structure [5] is dynamically created on constructing the different collaboration nodes of the e-Manufacturing scenario. In this structure a node in the scenario can be, at the same time, another collaboration scenario (MAS). This holarchy structure [6] is repeated until there is no need for further decomposition in the value chain. The collaborative scenario, which

\(^2\) [www.w3c.org/Submission/OWL-S/](http://www.w3c.org/Submission/OWL-S/)
Adriana Giret

implements the virtual enterprise, will execute and mediate among the virtual partners (agents) until the entire open manufacturing process is completed.

The system we have developed with the ANEMONA + JANUS approach, is designed to be hosted on a server or servers managed and owned by an organization of e-Manufacturing supported agents (eMSA). The eMSA agents, identified and specified using the ANEMONA guidelines, are the providers of collaboration services and security issues but not members of any of the open virtual enterprises that might use the environment. Apart from this group of supported agents, the system architecture provides a group of configurable agents, called collaboration-specific agents (CSA), which can be used by the enterprise virtual partners as mediator and facilitator agents in the collaborative scenarios. When instantiated, every CSA represents a capability enabler agent of a given virtual partner in the e-Manufacturing environment. This second group of agents are PROSA [12] types of agents (identified and specified using the PROSA Guidelines of ANEMONA) that can be used to configure any kind of open manufacturing process. In this way, we have four types of agents: product, resource, work-order, and staff. A product agent stores the process and product knowledge needed to insure the correct manufacture of the product. It acts as an information server for the other agents in the collaborative environment. A resource agent offers production capacity and functionality to the other agents. A work order agent represents a task in a manufacturing system. It is responsible for doing the work assigned on time and in the right way. It manages the products that are being produced, the product status models, and all the logistic processing information related to the task. A staff agent can assist the other types of agents with expert knowledge. Each of the CSAs is responsible for one of the following manufacturing controlling aspects: internal logistic, manufacturing planning, and resource management.

ASeM has two main components: a) support for the creation and management of collaborative business activities (to mediate services) in an open virtual enterprise, and; b) the use of teams of user configured agents that collaborate through service-oriented computing. The first is provided by the group of eMSA. While the second components, made up by groups of interacting CSAs, are created on-demand whenever a new open virtual enterprise is required. Moreover, the e-Manufacturing environment provides collaboration scenario patterns for CSAs that can be easily configured in order to setup a new virtual enterprise venture. The two components are well differentiated as conceptual layers that interact to support the e-Manufacturing environment. ASeM is designed to be executed on any JANUS platform.

Figure 6 shows the implementation of ASeM. The JANUS platform executes all the agents of ASeM as internal holons. Moreover, we have used the AgentWeb Gateway [19] for agent-service interface. Nevertheless, any FIPA compliant module for agent-service computing integration [20] could be used.

Figure 7 depicts the cooperation diagram for a virtual enterprise creation interaction. A registered manufacturing company starts the cooperation scenario defining the virtual enterprise features interacting with its Customer Mediator.
Fig. 6. The ASeM prototype

agent (using the Web-based interface). In this definition process, the virtual enterprise purpose is stated and the required Work Order Agents are defined. This data is used by the Customer Mediator agent in order to request the creation of the virtual enterprise to the OVE System Manager. When receiving such a request the OVE System Manager queries the Trust and Security Mediator in order to get trust data associated with the customer for the type of virtual enterprise that is being requested. Depending on the trust data, the OVE System Manager decides to create or not the virtual enterprise. In the first case the virtual enterprise is accepted and registered in the system, together with the list of Work Order Agents associated. In this process the services defined for the virtual enterprise are also registered by the OVE Service Manager. Finally the customer is informed on the creation of the virtual enterprise. When the virtual enterprise is refused, due to trust data, the Work Order Agents are deleted from the system and the customer is informed as well.

A manufacturing company can order its Customer Mediator agent to search for possibilities to joint a virtual enterprise. In this case the Customer Mediator agent starts a cooperation scenario with the Collaboration Mediator agent in order to keep updated information on the virtual enterprises that are advertising collaboration opportunities. When a collaboration opportunity is found the Collaboration Mediator facilitates all the data to the Customer Mediator who estimates the advantages to collaborate in. In case the Customer Mediator decides to collaborate, the Collaboration Mediator puts in contact the Customer Mediator with the Production Mediator agent and the Customer Mediator of the given virtual enterprise. All this process is supervised by the Trust and Security Mediator agent.
The manufacturing company is kept informed on the status of the different collaborations scenarios in which it is participating in thanks to the Distributed Workflow Manager, the OVE Status Monitor and its Customer Mediator agent. Moreover, the updated list of the work orders the manufacturing company has agreed to execute in its factory floor can be obtained.

6 Discussion and Final Remarks

The open virtual enterprise is an ongoing research field in which successful applications have began to appear. An interesting research topic has been the technology and modules needed [21–25] to support virtual enterprise processes which are based on interaction between services implemented by different organizations in the virtual enterprise. Some interesting research approaches include how to support service composition [26], dynamic service integration [22] and the management of service level agreements [22, 27].

Camarinha-Matos et al. proposed an open PRODNET architecture [28] to support industrial virtual enterprises with special focus on the needs of small and medium-sized enterprises. The architecture supports different enterprise behaviours through the internal module and the cooperation layer. Sandakly et al. [29] proposed an approach to build a virtual enterprise software infrastructure that offers persistence, concurrent access, coherence, and security on distributed data-store based on distributed shared memory paradigm.

Tang et al. developed a Web-based platform called E-DREAM [30] to support the distributed information management and role management in an agile virtual enterprise. Yoo and Kim [31] developed a Web-based knowledge manage-
ment system for facilitating seamless sharing of product data among application systems in virtual enterprises.

Despite the number of applications in the field, the majority of them where developed using a mix of different methods and tools. In same cases the engineers required to implement special modules to facilitate the interaction among the different tools used in order to assure compatibility. The ANEMONA + JANUS approach contributes to the field of Open Manufacturing System providing domain specific engineering guidelines and allowing a uniform development process in which the entire life cycle of the virtual enterprise is supported by a complete and robust development method. Moreover, the support for holonic structure execution and organization management provided by the JANUS platform facilitates the implementation activities and maintenance operations.

Right now, we are working on defining analysis and design patterns for Open Intelligent Manufacturing Systems. These patterns will be the starting point towards a Requirement-Guided development approach for Open Manufacturing Enterprise IS. In such an approach the development process will be a parameterizing process guided by Interaction patterns to incrementally build the system.

References