A FRAMEWORK FOR BUILDING DISTRIBUTED AGENT-BASED INDUSTRIAL APPLICATIONS

Dimitri Konstantas\(^1\), Ciarán Bryce\(^1\), Jean-Henry Morin\(^1\), Christian Kobel\(^3\)  
Apostolos Vontas\(^4\), Panos Hatzaras\(^4\), Adamantios Koumpis\(^4\)

\(^1\)Centre Universitaire d'Informatique, University of Geneva  
24 rue du General-Dufour CH-1211 Geneva 4, SWITZERLAND  
E-mail: [Dimitri.Konstantas][Ciaran.Bryce][Jean-Henry.Morin]@cui.unige.ch

\(^2\)University of Twente, Faculty of Informatics  
NL-7500 AE, Enschede, THE NETHERLANDS  
E-mail: dimitri@cs.utwente.nl

\(^3\)Pebble@ge S.A.  
7, place de la Fusterie, CH-1204 Geneva, SWITZERLAND  
E-Mail: kobel@pebbleage.ch

\(^4\)Unisoft S.A. Research Programmes Division  
Michael Kalou 6 GR 546 29 Thessaloniki, GREECE  
E-mail: [avontas][phatzaras][akoumpis]@the.unisoft.gr

Abstract: In the paper we present a fresh approach to viewing the technological and business and organizational aspects of agent-based systems utilized in industrial and manufacturing environments. Based on experience coming both from intensive research activities carried out by the authors’ organizations, and exposure to real market needs and technology utilization potential, our approach focuses on providing the essentials for the development, deployment, operation and evolution of agent-based technologies, infrastructures and (instances of such) systems. Policies, industrial practices and models are embedded in agents representing intra- and inter-system processes and activities. Copyright © 2001 IFAC

Keywords: Software engineering, Agents, Distributed models, Enterprise modeling, Enterprise integration

1. INTRODUCTION

At the technical level, the goal of our initiative is to develop an agent-based framework that is independent of any particular technology platform, mechanisms or the existing schools of thought. The platform must allow miscellaneous agent operation models and implementation environments can be integrated, like for example, workflow and information flow, process-activity models and ABC/ABM (Activity Base Costing/Activity Based Management).

However, we foresee, a turn from the still currently dominating paradigm of on-line Business Information Systems exchanging data and information using e.g. EDI as a technology means, to Real-Time Industrial Information Systems which rely through the appropriate modelling as well as layering of the Business Information Systems by means of e.g. using medatada facilities as presented by the MetaData Coalition Open Information Models.

Such a transition will affect the way we currently treat the processes of building and employing agent-based systems, especially in the manufacturing domain, as well as the way actors from both the academia and the industry are positioned with respect to the underlying economies and (technology) market development models.

Planning and administration of an agent-based system is a big issue in a current world of project-oriented, customer driven, "engineer-and-made-to-order" type in the software engineering discipline.

There are basically two agent communities. First, intelligent agents or multi-agents and second, mobile agents. The former rises from artificial intelligence and is focused on knowledge representation, collaboration, behaviour, avatars, etc. The later stemming from object-oriented, network and distributed system research address the issue of moving behaviour towards the source of data.
2. THE ADOPTED APPROACH AND RESEARCH AGENDA

According to the currently dominating practices, there should be two principal technological and business objectives in a project related to the introduction or customization of an agent-based system in an industrial application domain.

Following on the aforementioned case of employing agents for coping with the issue of production planning in the manufacturing environment, a first objective is to explore and enable time and cost efficient collaboration of the three key parties taking place in the process of manufacturing namely:

- **Software**, administrating and assisting the process of a working environment of the manufacturing domain e.g. an industrial shop-floor,
- **Hardware**, which plays an inevitable role e.g. in implementing a production plan, and,
- **Humans**, experienced and motivated human professionals, sharing their knowledge and expertise.

The second objective, taking into account current and emerging trends resulting from the wide adoption of the distributed computing paradigm, relates to "opening" of production planning from an internal, intra-enterprise activity, onto the Internet, by means of exploiting the global information infrastructure, which in this case concerns mainly all involved parties in the production planning activities (contractors, suppliers, customers), that are part of distributed (Intranet / Extranet / Internet) working environment.

At a technological level, the principal motivation for the community of ERP vendors in such projects is to replace a "traditional" process of production planning (which is currently based on database manipulation and fixed product and process data (order) exchange among members of a supply chain) by the process of communication, bidding, negotiation, knowledge manipulation and requirements delegation within a community of agents, i.e. actors involved in the whole process of production planning.

In the context of the above, by agent we mean the following entities:

- **software-bot**, which is an integrated software system such as a conventional business information system (e.g. Unisof’s ATLANTIS ERP series) or its part, knowledge based system, Web based application providing/pushing information, a specific internet search engine, etc.,
- **hardware-bot**, which is an ip-addressable PLC machine, CNC machine, sensor, etc.,
- **human-bot**, who is an expert, user, decision-maker or any multi-skilled highly motivated personnel involved in the production planning process.

In any of the adopted approaches, the adoption of the above triad of agents facilitates the investments made in designing and implementing agent-based interaction environments where (business) processes are represented by the three agent classes, being capable to "encapsulate" the policies, business practices and models of different business activities like for example – in the case of an ERP implementation - order management, payroll, warehouse and stock control. By combining different agents offering diverse services we will be able to compose complex applications for monitoring and managing in real time networks of extended enterprises, which may be networks of suppliers and providers, clusters of contracting companies, Plug-In Enterprises, etc. According to this scenario, integration of value-added components, in the form of e.g. software or hardware agents, from the different participating entities may be seamlessly achieved.

Today business systems, like SAP R3, provide integrated solutions for companies. Any new extensions/adaptations of the ISs must be either implemented by the IS designer or supported by (narrow scope) access to IS functionality by means, for example, of ODBC or native drivers. In this way there is little space for SMEs to provide highly customised value added components reflecting the particularities of the specific end user when these do not conform with the base IS.

Our vision is to provide a framework in which the various technological and scientific approaches will be enabled to interoperate and capitalise on this aspect, where even SME IT actors will be able to provide self contained software business components in the form of agents that will by seamlessly integrated to create an IS tailored to the needs of the specific end-user, as this is today possible for PC hardware.

Such a framework has to be compatible with both agent community approaches as well as with the existing IS solutions, and can be introduced in a company in an incremental way.

The appropriate software gateways will have to be easily designed and implemented allowing the communication of existing systems of both approaches. Designing proxy agents that represent agent communities to each other and allow intercommunication will do this.

As a result an existing (operational) agent-based IS can be slowly replaced or augmented with the integration of new specialised composite agent components which may be constructed by different business software vendors.
3. BACKGROUND OF THE ACTIVE INFORMATION ENTITIES

Active Information Entities (AIEs) come as an evolution of research work reported in (Morin et al., 1999), as it has evolved in order to reflect the complexity and the idiosyncratic elements of the application domain in industrial enterprises. AIEs should thus be regarded as an operational extension of the notion of the Active Business Object (ABO).

The objective here is to support industrial enterprises capitalise on the benefits of the information society, by enabling the adoption of a new paradigm: instead of reactively responding to needs for information and data exchange amongst intra- and inter-enterprise structures (e.g. networks of central, regional or local government bodies and departments). AIEs provide an active infrastructure based on agent technology that will realise an open environment integrating components from various specialised IT vendor solutions.

Specifically, we propose building and integrating of IS from software components based on a scaleable modeling approach that supports the exchange of business data and information encapsulated in agents (i.e. the Active Information Entities). These agents include "methods" that allow them to act and interact in an independent way according to the needs and the operational requirements of the activity in which they participate.

Based on experience with research activities of the OSG group at the University of Geneva (Konstantas et al., 1995 & 1996), and on Unisoft's exposure to real market needs and technology potential, ABO focuses on providing essential technology and infrastructure for the development, deployment, operation and evolution of business systems. Policies, business practices and models are embedded in agents representing intra- and inter-enterprise processes and activities.

At the technical level, the goal of AIE is to develop an agent-based support framework upon which different business operation models and environments can be implemented and integrated, like for example, workflow and information flow, process-activity models and ABC/ABM (Activity Base Costing/Activity Based Management), see also (Morin et al., 1999) and in (Johnson et al., 1987). The basic building block of AIE is the Mobile Agent: a self contained reusable active entity encapsulating data and code and implementing specific services of the particular IS application.

Mobile agents (White, 1996) (on which we base our AIE concept) is now a recognised programming method for distributed working environments (Intranet/Extranet/Internet). By allowing for the migration of computation between hosts and platforms, networked services can be made more flexible and cost effective. In AIE we design and implement a framework augmenting existing agent deployment platforms with tools for implementing, monitoring and managing business processes (i.e. processes for work- and information-flow) for an extended enterprise environment. This is an environment made up of departments or companies distributed over the network. The support framework enables companies / departments to join and leave the business flow, offer services for a work- or information-flow process, and have these services accounted for their resource utilisation.

The main idea is to design and develop an agent platform where mobile agents represent business processes. These agents encapsulate the policies, business practices and models of different business activities like for example order management, payroll, warehouse and stock control. By combining different agents offering diverse services we will be able to compose complex applications for monitoring and managing in real time an extended enterprise. According to this scenario, integration of value-added component, in the form of agents, from different SMEs will be seamlessly achieved.

To give a solid example of the AIE vision for industrial enterprises, one can compare it to the wide spread and evolution of PCs. Today a PC is composed from different hardware components developed by different companies. Each company excels in the development of a specific hardware component, like for example a CPU, a motherboard, memory, special purpose cards etc. As a result a wide range of solutions is available on the market and users can easily compose PCs to fit their specific needs and preferences. At the same time SMEs can bring to the market highly specialised hardware products addressing niche markets and more harmonised life cycles.

Today business systems, like SAP R3, provide integrated solutions for companies. Any new extensions/adaptations of the ISs must be either implemented by the IS designer or supported by (narrow scope) access to IS functionality by means, for example, of ODBC or native drivers. In this way there is little space for SMEs to provide highly customised value added components reflecting the particularities of the specific end user when these do not conform to the base IS. AIE’s vision is to provide a technology where various levels of a national, regional or local industrial / manufacturing infrastructure will be able to provide self contained software components in the form of agents that will by seamlessly integrated to create an IS tailored to the needs of the specific end-user, as this is today possible for PC hardware.

The proposed technology is compatible with existing IS solutions and can be introduced in a company in an incremental way. Such software gateways can be easily designed and implemented allowing the communication of existing ISs with the AIE platform.
as this will emerge as implementation of the proposed information architecture for industrial enterprises. At a technical level, this will be enabled by designing proxy agents that will represent the existing IS in the AIE platform and allow intercommunication. As a result the existing ISs can be slowly replaced or augmented with the integration of new specialised composite AIEs constructed by different software vendors.

4. DESIGN ISSUES AND GOALS

The design issues under investigation for AIEs are divided into two general areas. The first deals with the necessary extensions to the mobile agent infrastructure while the second is concerned with looking at how agents (active objects) can help structure the existing process backbone of business IS.

4.1. Developing a Mobile Agent Infrastructure

To make agent technology a more general support infrastructure, enabling capitalisation by industrial enterprises, the following design issues are investigated:

- **Programming models:** An agent sent out onto the net within a virtual process must be able to react to a number of events, ranging from network failures, broken pointer links to information and security attacks, to time-outs for the delivery of their content and spawning and controlling "children" agents for handling tasks in parallel. It is not yet clear what might be the most convenient model of programming agents to handle such events.

- **Backbone Integration:** A real electronic commerce backbone needs to marry the benefits of agent technology with standard middleware technology. Integration of the AIE agent platform with a middleware architecture such as CORBA or DCOM is an important goal of AIE (Bryce, 1999).

- **Resource control:** Agents must be made accountable for the resources that they consume at a location during a process. To do this, the identity of the enterprise that created the agent and which should be charged for a given resource consumption must be easily established. Efficient mechanisms for achieving this will be designed, based on existing standards and technologies.

4.2. Agents and Process Backbone

The second issue that needs to be addressed for the ABO platform design is how the agents can be exploited to alternatively structure the business flow process and reservation protocols. For instance, agents can be used to remotely configure resources at a site; this is useful for improving the monitoring aspect of work- or information-flow processes.

In particular, agent support will be implemented fulfilling the following requirements:

- **Security:** agents allow computations to be moved. From a security viewpoint, this can be used to move process computations to safe areas to avoid certain attacks (Vitek, 1997).

- **Off-line operation:** to reduce the cost of using the network, it is important to identify those services in a work-flow process that can be down-loaded to a client site in the form of agents.

- **Reliability:** bad network connections can also be overcome by computation mobility.

The integrated outcome of such an approach is a coherent software framework on top of which software components for networked enterprises can be developed and seamlessly integrated and deployed in a small-, medium- and large-scale industrial enterprises, similarly to the case of tackling with a small-, medium- or large-scale commercial enterprise environment.

In this context, we address issues related to component-based software engineering, as we adopt a component-based approach, focusing on component integration and the concept of evolutionary re-configuration. AIEs as approached may transform the organisational and functional levels of an industrial enterprise, thus enabling the smooth transition to more efficient and cost-effective ways of structuring work both at the intra- and at the inter-enterprise level. As they will be coming from different sources (i.e. ERP systems of the different industrial enterprises) may thus form the "cornerstones" of new, complex systems and services, thus taking advantage of the "system families" concepts and stimulating both real-life practice improvement and the take-up of the associated software technologies.

During the last 20 years Information System (IS) design and implementation has evolved from the monolithic flat programming (e.g. Cobol based), to modular programming (e.g. C), to object-oriented programming (e.g. C++) and finally today in agent based programming (e.g. Java). Each evolutionary step improved the ability of component reuse in the design and implementation of applications and introduced a new paradigm for application and component interoperation.

With monolithic programming the reuse of components for the implementation of applications was near zero and the application interoperation was performed with raw unstructured data. Modular programming allowed the design of applications based on well defined module interfaces which were able to be reused by different applications and introduced the notion of interfaces (API and RPC) and client-server model for the interoperation of applications.

Object-oriented programming increased component reusability introducing the notion of objects encapsulating data and behavior, and proving application interoperation via the exchange of messages. Object libraries provided the required
mechanisms for the implementation of applications. A further evolutionary step was the introduction of middleware (e.g. CORBA) that allowed the composition of applications based on the notion of services. Complex applications were designed by decomposition in required services which were to be found on the network.

The basic characteristic of the above application design and implementation models is that they are based on the client-server model where the client sends data to the server who process them and returns them to the client.

During the last few years a new model for the design and implementation of applications has appeared: the agent based application programming. Agents allow application interoperability with the exchange of code, that is, agents, that encapsulate a specific behavior. With agent programming instead of sending the data to the location of the behavior, we import the behavior at the location of the data.

However without realizing it, the agent-programming model has taken us one step back in the design and implementation of applications, when compared with the client-server evolution model approaches. That is, whereas middleware was an evolutionary step from OO programming allowing us to implement applications in terms of high level services instead of behavior, agent based programming has brought us back to application design based on behavior.

The first contribution of AIEs is providing a framework that will allow application design and implementation based on the notion of services’ composition, facilitating component integration not in terms of API and network middleware standardization, but with open-ended mobile agents. The AIE framework platform that will enable customized tailored composition of IS for Pas according to the devised information architecture, where AIE-based applications will interoperate using active objects (agents) which encapsulate services implementing policies and practices of applications.

Based on this platform design and implementation of the AIE concept will take place, proving innovative applications introducing agent technology for the management and control of processes and activities. Implementation of an information architecture based on the AIE concept will promote the employment of agents in industrial information systems by means of using a consistent and integrated approach that enables policies, business practices and models to be embedded in intra- and inter-enterprise processes and activities.

5. EXAMPLE APPLICATIONS

In order to test and verify the ideas and concepts two customisations of the proposed environment will provide an example of the application layer. The adopted framework may be employed in order to provide a common abstraction to highly differentiated industrial enterprise environments.

More specifically, the ABC/M model relies on a producer-consumer relationship between three concepts that are necessary and sufficient to describe and model an industrial environment: processes, activities and resources. We provide below a definition for each:

- **Process**: a process is a notion freed from any organizational or structural dimension. It captures a functional dimension, transverse to the organization, oriented towards a production objective. A process describes a functional objective of the business and is composed of activities and/or sub-processes together with the necessary resources involved in their fulfillment.

- **Activity**: an activity identifies an action which can be characterized by a verb and an object upon which the action applies. Activities are necessary in (i.e., consumed) fulfilling processes. Moreover, activities consume resources necessary to their fulfillment.

- **Resource**: a resource identifies in a very broad sense any production factor whose consumption occurs within activities (i.e., which are necessary for fulfilling activities).

We have decomposed the ABC/ABM AIEs in two distinct categories: Referential AIEs and Operational ABOs. The first category serving the purpose of capturing the “business” in terms of a related model. The second category, the operational AIEs to capture the actual life (instances) of the business processes, through the corresponding information entities.

A central issue for ABC/ABC in the particular application context is to be able at all time to measure everything and hence report activity through management instrument panels and allow action on the “business” to be taken both at the referential level and at the operational level.

Although both categories will be modeled as agents, only the operational AIE agents will have migration capabilities. Referential AIE agents do not need mobility as they are bound to a given abstraction describing a business reason for which we consider them as static agents.

From there on, it becomes possible to measure and monitor these agents in a way similar to placing probes within them. As a result, managers, policy- and decision-makers can be given tools allowing them to build management instrument panels showing in real time the indicators for which they have expressed interest. These indicators can reflect information from either the referential level (i.e., Referential AIEs) or from the operational level (i.e., Operational AIEs).
Furthermore, it also becomes possible to browse and navigate through the AIEs to audit the business trying to identify poor business patterns and malfunctions and thus take corresponding actions. Finally, goal oriented simulations and scenario evaluation become possible.

REFERENCES

Konstantas Dimitri, "Migration of Legacy Applications to a CORBA Platform: a Case Study," Proceedings of the IFIP/IEEE International Conference on Distributed Platforms - ICDP'96, Dresden, Germany, February 27 - March 1, 1996