The multi agent paradigm and organizational abstractions in construction e-business

E. Obonyo  
*University of Florida, USA*

Chimay Anumba  
*The Pennsylvania State University, USA*

Abstract

The agent paradigm has been used by several researchers to build several complex construction industry specific systems ranging from e-business applications, cross-disciplinary communication to supply-chain management to contract administration. Because of the slow progress in the maturity of the deployment infrastructure, the implemented agent-based applications are generally based on models for closed systems. Given that recent developments in agent-oriented software engineering allow the definition of models that exploit the use of organizational abstractions, there is need to redefine classical agent-systems using the organizational metaphor. This paper focuses on Construction e-business as a deployment context. It starts with background information in the domain and proceeds to describe classical agent-centered models that were used to develop APRON, a construction e-business prototype. The paper then discusses the main drawbacks of the agent-centered approach and demonstrates how the organizational metaphor can be used to address these limitations providing an exemplary procurement scenario.

*Keywords*: agent-centered, organizational-centered, models, construction e-business

1 Introduction

The Construction Industry embodies professionals in information-intensive activities. Decisions have to be made based on the information available, but there are heterogeneous, distributed, dynamic, semi-structured and unstructured data and knowledge sources. The existing information sources have an open architecture with structures exhibiting dynamism. The components change and cannot be predetermined because they are implemented by different people, at different times, using different tools and different techniques. This problem can now be rigorously addressed using emerging paradigms in information technology that facilitate the modelling of the fragmentation in the construction industry. Distributed Artificial Intelligence (typically implemented in the form of intelligent agents) offers an innovative approach to overcoming this problem. The phrase “intelligent agents” is used here to refer to systems capable of autonomous, purposeful action in the real world whereas “multi-agent system” (MAS) refers to a computational system in which two or more (homogenous or heterogeneous) agents interact or work together to perform a set of tasks or to satisfy a set of goals (Lesser, 1999). Such a system comprises (1) an environment, (2) a set of passive objects that can be associated with a position in the environment, (3) an assembly of agents, which are specific objects representing active entities of the system, (4) an assembly of relations linking objects (and thus agents), (5) an assembly of operations with which agents perceive, produce, consume,
transform and manipulate objects and (6) operators representing the assembly as well as reaction modifications (Ferber, 1999). A multi-agent system is therefore a consolidation of autonomous “problem solvers.”

The notion of a community of agents cooperating to fulfill a complex task is the fundamental benefit of deploying the agent-based technologies in a system. Interestingly, most of the existing applications are generally based on models for closed systems thereby limiting the potential for exploiting agents in systems deployed by different developers. This can be largely attributed to the fact that agent technology is a relatively new field and the enabling infrastructure is still maturing. In the subsequent sections proceeds to provide an overview of the classical agent-centered models that were used to develop APRON, a construction e-business prototype. The paper then discusses the main drawbacks of the agent-centered approach and demonstrates how the organizational metaphor can be used to address these limitations providing an exemplary procurement scenario.

2 Deploying agent-centered multi-agent system

APRON, a construction e-business prototype was deployed to demonstrate the potential of using the agent paradigm to enhance information management. The simulated use case in the implemented prototype was based on processing product information for the specification and procurement of light bulbs from the Philips Lighting Web site (URL1). The site hosts close to 200 catalogues in Adobe Acrobat PDF format. The information that would be of interest to an end-user such as wattage, cap size and voltage is presented in a semi-structured format. The Web site does not have a search facility that would support guided navigation based on attributes such as wattage and voltage. It is also not possible to query the information directly from any another application. Furthermore, relevant data has to be re-keyed for reuse in a different application. The MaSE approach was used to design this prototype. MaSE is a further abstraction of the object-oriented paradigm where agents are a specialization of objects (DeLoach et al., 2004a). Unlike simple objects whose methods can be evoked by other objects, agents in this framework coordinate with each other via conversations and act proactively to accomplish individual and system-wide goals. Further details on agent classes, agent conversations and the resulting MAS architecture can be found in Obonyo 2004, Obonyo et al. 2004, 2005a and 2005b.

An important parameter in the definition of agent-based systems is their reliance on a society of agents interacting and cooperating to achieve some collective goals. Much of the pioneer research work in this paradigm has dwelt on deploying ‘agent-centered multi-agent systems’ (ACMAS) such as APRON. ACMAS-driven research focuses on the internal mental state of an agent the relationship between these states and its overall behaviour. Ferber et al. (2004) pointed out that in this view communications become speech acts whose meaning may be described in terms of the mental states of an agent as is evident in agent communication languages such as the KQML and FIPA ACL. This has resulted in the design and development of agent-based systems in which agents can only communicate with one another in a closed system. Because agents generally exist in the context of multi-agent software systems with some defined global behaviour being derived from the interaction of constituent agents, having a closed system greatly undermines the potential benefits of cooperation (Zambonelli et al., 2001).

The main benefits of defining a societal structure for MAS include reducing the system’s complexity, increasing the system’s efficiency, and enabling more accurately model the problem being tackled (Jennings and Wooldridge, 2000). Without this societal structure the patterns and the outcomes of the interactions are inherently unpredictable and predicting the behaviour of the overall system based on its constituent components is extremely difficult (sometimes impossible) because of the high likelihood of emergent (and unwanted) behaviour (Jennings, 2000). As Ferber et al. (2004) points out, in such a scenario it is not possible for agents designed by different designers to interact
unless one makes some assumptions about the primitives of communications and the architecture of agents. As agents based on the ACMAS lack have access to these constraints, they are constrained to using the same language and have to be built using very similar architectures. Moreover, such an approach lacking generality and being tuned to specific systems and agent architectures exploits abstractions that are unsuitable for modeling agent-based systems (Bussmann, 1998; Ferber and Gutknecht, 1998; Ferber et al., 2004).

There are three other major weaknesses of the ACMAS model (Ferber et al. 2004). Firstly, since all agents’ communication is without any external control, an applications designer often has to balance between: 1) allowing free interaction of agents thus making it easy for an agent to act as a pirate and use the system fraudulently, and; 2) implementing too strong security measures that could prevent the system from working efficiently in domains where speed and response is of critical concern. Secondly, with AOMAS all agents are accessible from everywhere, grouping entities that work closely together into “packages” that may or may not be hidden as is done in software engineering. The actual challenge within a MAS model would be coming up with a dynamic framework for grouping agents that work together. Finally, since in the ACMAS model the platform is the only supported framework, it is not possible to fully exploit the component concept that is used in classical engineering as an abstract architecture.

3 Organizational-centered MAS and construction e-business

There has been a emerging interest in the use of macro-level concepts such as ‘organizations’, ‘groups’, ‘communities’, ‘roles’ in designing multi-agent systems as a possible solutions to the limitation of ACMAS models. Ferber et al. (2004) has shown that the OCMAS approach can altogether dispense with the use of mental states used in classical ACMAS models. An agent-based system such as the one designed for construction e-business can actually be viewed as several interacting organizations and it is in fact possible for an agent to be part of multiple organizations (Zambonelli et al., 2001). It is therefore not surprising that organizational constructs have been generally perceived as being the first-class entities in MAS. Such agent-based systems can thus be deemed to have computational mechanisms for flexibly forming, maintaining and disbanding organizations. This implies that the notion of a primitive component can be varied according to the needs of the observer. Additionally, such structures provide a variety of stable intermediate forms in which individual agents or organizational groupings can be developed in relative isolation and then added into the system in an incremental manner (Jennings, 2000). Since the behaviour of agents within a system is based on the behaviour and structure of human organizations, each agent has a clearly defined role within the system. Consequently, interactions are no longer mere expressions of classical object-oriented, interdependencies (Booch, 1994), but are also a characterization of the position occupied by an agent within the organization. The organizational metaphor also simplifies the design of the system by separating the component level (intra-agent) design dimensions from the system level (inter-agent). The organizational metaphor also makes it easier for the designed MAS to closely mirror the real-world organizations they are intended to support (Zambonelli et al., 2001).

To explore the OCMAS notion further, an exemplary architecture extending the implemented APRON prototype was designed leveraging on the Agent/Group/Role (AGR) model (Ferber et al., 2004; Ferber and Gutknecht, 1998). In the AGR model an agent is only specified as an active communicating entity playing (multiple) roles within (several) groups while organisational concepts such as groups, roles, structures and dependencies, are first-class citizens of a design methodology. A key attribute of the AGR model is its minimalist structure-based view of organizations as a role-group structure imposed on the agents. AGR also says that agents can have their joint behaviour orchestrated by interaction protocols, but the nature and the primitives to describe such protocols are left open. Based a simple use case such as the one shown in Figure 1 the AGR model can be used to model a
MAS organizational structure for Construction e-business with interactions between three group structures: a procurer group structure (ProcurerGS); the manufacturers group structure (ProviderGS), and a contracts group structure (ContractGS) used when a client decides to buy a product from the provider.

The “Broker” is the same agent with dual roles in the client group and the manufacturer group. When an agent enters a client group, the client asks the broker for a product. Then the broker sends a call for proposal to manufacturers. The resulting proposals are presented to the client for product selection. In case of a suitable match, a contract group is created with both the client and the chosen manufacturer, taking the respective role of “Buyer” and “Seller.” This process can be repeated for all the other organizational concepts including “organization rules” and “organization patterns.”

4 Discussion

Before agent-based systems can be deployed in large-scale real life applications within the context of construction e-business, it is necessary for researchers to reduce the gap between the real world and the design models used in agent-oriented software engineering. In particular, MAS need to be designed for the deployment of open systems that can truly exploit the flexibility and autonomy inherent in intelligent agents. The work presented in this paper provides a conceptual framework for extending an agent-centered prototype for the specification and procurement of construction products. Further details on the implementation and evaluation of the deployed proof-of-concept (APRON) have been provided elsewhere (Obonyo 2004, Obonyo et al. 2004, 2005a and 2005b). The focus of this paper was establishing the need to redesign the application using organization-centered abstractions as a way of enabling the system to interact with agents developed by other people thus truly exploiting the full benefits of the agent paradigm.
The main limitation of the APRON prototype that was originally implemented was the fact that it was not an open system. This limitation applies to most (if not all) other agent-based construction applications. Deployment of open agent-based applications for construction e-business has not been possible largely because the existing design and development platforms focus on the internal mental states of an agent, the relation between these states and the agents overall behavior using agent communication languages such as the KQML and FIPA ACL. Consequently, the resulting applications are based on agent-centered multi-agent systems in which agents can only communicate with one another in a closed system. This paper has explored the use of the organizational metaphor to encapsulate macro concepts in the design of MAS for deploying applications that are more open and dynamic. There has been a growing interest among researchers in agent-oriented software engineering in developing methodologies for modelling the organizational abstractions. Therefore, the redesign of ARON into an OCMAS model leverages on developments in the agent infrastructure.

The paper has shown examples of organizational abstractions for construction e-business. Defining open and dynamic models implies that the agents roaming on the Web will inevitably join groups designed by other developers using different terms. As the organizational metaphor becomes truly global and models interaction between agents in different groups implemented by different developers, there will be a need for reconciling differences in the use of concepts and terms. Before the organizational metaphor can take root as a superior approach to analyzing and designing agent-based system, the agent community will have to address the semantic complexities inherent in the use of the OCMAS model. This problem can be addressed through defining an ontology for the MAS organizational concepts. Preliminary reviews of emerging trends in this area has revealed that an example of such ontology has been implement in a different domain by Coutinho et al. (2005). Further research will involve extending this ontology for use in defining an OCMAS model for construction e-business.

References

BOOCH, G. 1994, Object-oriented analysis and design (second edition), Addison Wesley, Reading, MA.


