Chapter XXII
On the Design of Multiagent, Context-Aware, and Mobile Systems

Tagelsir Mohamed Gasmelseid
King Faisal University, Saudi Arabia

ABSTRACT

This chapter addresses the software engineering dimensions associated with the development of mobile and context-aware multiagent systems. It argues that despite the growing deployment of such systems in different application domains little has been done with regards to their analysis and design methodologies. The author argues that the introduction of mobility and context awareness raises three main challenges that deserve a paradigm shift: the challenge of information integrity, service availability on mobile devices, and the complexity of decision modeling. Because they reflect different operational and procedural dimensions, the author argues that the conventional software engineering practices used with intelligent systems that possess other agency qualities need to be "re-engineered." The chapter emphasizes that the envisioned methodology should reflect a thorough understanding of decision environments, domains representation, and organizational and decision-making structures. Furthermore, the chapter provides a description for the appropriate enablers necessary for integrated implementation.

INTRODUCTION

The significant advances exhibited in the field of mobile and wireless information systems have resulted in a rapid proliferation of mobile information devices and considerable improvement in their capabilities. Devices such as cellular or Web phones, laptop computers, and personal digital assistants (PDA) have developed from basic means of communication and simple electronic calendar management units into computing devices capable of transmitting and processing data with wireless access to the Internet. Their uses in large scale distributed networks and grids allow new distributed applications to emerge in different domains. They have been widely used in e-business, e-learning, supply chain management, virtual enterprises (Jain, Aparicio, & Singh, 1999), information retrieval (Cabri, Leonardi, & Zambonelli, 2000), Internet-based auctions (Sandholm & Huai, 2000), distributed network management (Du, Li, & Chang, 2003), resource management and broadband intelligent networks (Chatzipapadopoulos, Perdikeas, & Venieris, 2000), telecommunication services, and mobile and wireless computing *(Keng, Ee-Peng, & Zixing,* 2001). They have enhanced the functionality of such systems by improving the availability, accessibility, and management of decentralized repositories of concurrent data. In addition, the transformation of business environments and availability of mobile technology have encouraged the study of "context awareness" as a leading feature. Context awareness is directly and inexorably related to "mobility" where decision makers and users of mobile systems can use mobile networks to access data and carryout transactions using a wide range of intelligent units and applications.

MULTIAGENT MOBILE AND CONTEXT AWARE SYSTEMS

An intelligent agent is an autonomous, computational software entity that has access to one or more heterogeneous and geographically distributed information sources, and that proactively acquires, mediates, and maintains relevant information (Gasmelseid, in press). Intelligent agents are used to carry out functions on behalf of their users, other agents, or programs with some degree of autonomy using multiple information and communication platforms. While some agents have been used for information search and retrieval and the management of information overload, others perform repetitive activities and specific tasks such as scheduling and interface presentation, task delegation, users training, event monitoring, information search, matchmaking, and filtering (Wooldridge & Jennings, 1995; Gasmelseid, 2007).

Within the context of a distributed work environment, software agents can assist in facilitating coordination, cooperation, and interaction among different agencies (Gasmelseid, 2007). Their deployment in electronic commerce, traffic control, healthcare provisioning, portfolio management, and telecommunications proved their relevance to handle complex, distributed problems involving a multiplicity of interconnected processes whose solutions demand the allocation of fusion of information and expertise from demographically distributed sources (Gasmelseid, 2006). Using an agent in the healthcare sector, for example, patients can receive remote monitoring and telemedicine services from one or more healthcare service providers (i.e., hospitals, physicians, nurses, pharmacies, laboratories, clinics, emergency centers, and consultants) in a high quality cost effective form. In a distributed healthcare setting, general practitioners, hospital specialists, nurses, and home care organizations have to work together to provide the appropriate care to a sick patient (Huang et al, 1995). Industrial applications of agent technology, on the other hand, include the application of contract net task allocation protocol in a manufacturing environment. In process control, multiagent systems have been applied in electricity transportation management, and particle accelerator control (Jennings & Wooldridge, 1998).

In response to the growing situational and business complexities and the increasing focus on resource sharing, agents are usually assembled into "multiagent organizations" and structures rather than being used as individual components. Due to the lack of an universally accepted definition for the concepts that govern and dictate "agency," multiagent systems are described in accordance with their functions based on the qualities possessed by the agents making them up.

The transformations exhibited in business environments, the growing importance of "delegation" and resource sharing as a business imperative, and the improvement of technological platforms have moved "mobility" and "context awareness" to the list of highly acknowledged agent qualities. While such qualities are not usually considered on individual bases, they are coupled with other qualities in pursuit of orchestrating and harmonizing the functionality of the entire multiagent organization. Therefore, when mobile agents are used to constitute the entire multiagent system they can jointly use knowledge and resources to solve problems in a context-dependent way (Gasmelseid, 2006a) and move interactively to realize some objectives and perform functions.

In telecommunications terms, mobility is defined as the ability to "wirelessly" access all of the services that one would normally have in a fixed wired line environment such as a home or office, from anywhere. It includes terminal mobility (provided by wireless access), personal mobility (based on personal numbers), and service portability supported by the capabilities of intelligent networks. Especially in technology-intensive and information-rich distributed environments, mobile agents interact to gather information, route processing outcomes, update a database, and read performance levels, among others. In addition to its ability to meet objectives, the efficiency of any mobile multiagent system is also based on its capacity to adapt to changing environments.

Despite the increasing deployment of mobile agent-based systems in different domains (e.g., education, medicine, airlines, commerce, etc.), their use is being constrained by the unpredictable variation in network quality, lowered trust and robustness of mobile elements, limitations on local resources imposed by weight and size constraints, and concern for battery and power consumption (Satyanarayanan, 2001). Fortunately, the unprecedented advancement exhibited in the fields of mobile networking and performance-improving techniques, consistency of mobile information access, adaptivity and location sensitivity, and energy saving are relaxing these constraints. These developments are directed, among others, towards fostering adaptability and ensuring a balance between autonomy (reliance on static servers dictated by the relative resource poverty of mobile units and their lower trust and robustness) and interdependence (self-reliance necessary to cope with unreliable and low-performance networks, and power consumption). The use of mobile agents demands unique mobile hardware and software components because the entire multiagent system exhibits considerable data management complexities originating from the mobility of devices, hosts (and connections among them), and users on the one hand and the unique way mobile components operate on a wireless network and in disconnected mode on the other hand.

In addition to mobility, issues related to context awareness are also emerging. The importance of bringing "context awareness" into the decision-making environment, especially for global enterprises, is dictated by a wide range of factors:

a. The growing level of complexity, sophistication, and downside risks associated with distributed (yet integrated) business transactions.
b. The growing emphasis on accountability, corporate responsibility, and orchestrated processing and the increasing importance of real time information. The fear of taking decisions based on incomplete information (using cost benefit analysis and institutional trade offs) increased the importance of acquiring different types of information and introduced the question of "context awareness" into the "mobile" decision-making domain.

Context awareness relates to the ability of the entire "system" to sense and use the elements of its surrounding environment. A mobile context-aware multiagent system appreciates its environment and adapts its configuration and functionalities in accordance with available context information that can be used to characterize the situation of any entity such as the location of the user, collection of nearby objects and people, hosts, accessible devices, and change to them over time. While using context information and context-aware systems in physical applications (like robots) may be simple, understanding, acquiring, and using context information in decision support domains is a complex task. The complexity stems from the difficulty of using and exploiting new classes of context-aware applications in a changing technical and operational environment in which different types of context information is generated. However, despite the production of different types of context information, what constitutes "context-relevance" of information in a decision-making domain is contingent upon:

1. The nature and magnitude of the potential change and/or improvement an information element generates about the states of decision making, for example, minimizing risk or moving to the state of "certainty."
2. The degree of information integrity to be incorporated in the decision-making environment and the variables interacting across it.

Within this context, mobile and multiagent systems are being widely used for mobility management and enhancement of context awareness. Their use tends to allow for better performance, scalability, portability, connectivity, bandwidth, energy, and robustness (*Ee-Peng & Keng,* 2001; Yu Jiao & Hurson, 2004). They assist in balancing workload because, by migrating from the mobile device to the core network, agents can take full advantage of the high bandwidth of the wired portion of the network and the high computation capability of servers and workstations. They also assist in reducing network traffic because the migration capability of mobile agents allows them to handle tasks locally instead of passing messages between the involved databases.

The ability of a multiagent context-aware system to provide decision support by sensing, extracting, interpreting, and using decision-related context information is based on the following considerations:

1. The ability of system developers to model the decision-making situation and configure it in a multiagent mobile and context-aware setup. This process demands the articulation of context-related information that relate to specific decision-making patterns (individual vs. group) and states (i.e., risk, certainty, uncertainty).
2. Modeling the magnitude of change associated with context information to improve its candidacy of capturing; that is, modeling data to crystallize the types of context-related information can be regarded as a "perfect information" and whose inclusion in multiagent, mobile, and context-aware decision-making processes makes a difference in the states of decision making and accordingly, the patterns adopted. In general terms, contextual knowledge associated with the current focus of attention (contextual knowledge) can be extracted, assembled, and structured to form the proceduralized context that is used in the current focus as a "chunk of knowledge." The part of context information which is not relevant for current focus or operations can be regarded as external knowledge.
3. Analyzing the orientation of decision makers and the way such orientations affect the nature and magnitude of context information and its utilization.

In the absence of concrete mobile and context-aware software engineering methodologies oriented towards decision-making environments, mobility and context awareness presents three types of uncertainties. First, information held in a mobile device is likely to be incomplete or outdated and may not reliably support user's needs in critical situations. Second, the availability of services over mobile and wireless networks necessary for business transactions varies in accordance with network traffic, the change of locations of users and machines, and the type and size of mobile devices. Third, modeling the behavior of decision makers is a prerequisite for the articulation and modeling of context-aware applications, a requirement that is difficult to achieve.

AGENT ORIENTED SOFTWARE ENGINEERING (AOSE) REVISITED

The recent technological developments have been accompanied with a shift from "conventional" to "agent-oriented" software engineering methodologies that can be used for engineering the software that has the concept of agents as its core computational abstraction particularly in the context of complex, open, networked, large, and heterogeneous applications (Gerhard, 2002; Jennings, 2001; Jennings & Wooldridge, 2002; Wei, 2001). Such migration is being motivated by the requirement that any software life cycle, process, or product model must be tailored towards the needs of the application domain of the target system (Basili et al, 1994). The main purposes of AOSE are to create methodologies and tools that enable inexpensive development and maintenance of agent-based software in a flexible, easy-to-use, scalable, and quality fashion (Erol, Lang, & Levy, 2000). Another purpose is to improve the ability of developers to construct flexible systems with complex and sophisticated behavior by combining highly modular components in the form of a multiagent organization. The adoption of AOSE methodologies to emphasize the underpinning organizational context to agents' interactions is seen as a vehicle for managing "complexities" through decomposition (dividing problems into manageable chunks in order to limit the designer's scope), abstraction (defining a simplified model of the system to emphasize and supports its properties), and organization (defining and managing the interrelationships between the various problem-solving components (Jennings, 2001). Based on this understanding, adopting an agent-oriented approach, "designers" can use multiple agents to represent the decentralized nature of problems and the multiplicity of control which incorporates different perspectives and/or competing interests.

Although many object-oriented analyses view the world as a set of autonomous agents that collaborate to perform some higher level function based on localization and encapsulation considerations (Booch, 1994; Parunak, 1999), agents can follow the same trend by localizing purpose inside each agent, giving each agent its own thread of control, and encapsulating action selection to support usability and system integration. However, despite the wide spreading and deployment of agent-oriented applications, Wooldridge and Jennings (1999) argue that agent-oriented software engineering methods may have some political, conceptual, analysis and design, agent-level, and society-level pitfalls. While political pitfalls occur when the concept of agents is oversold or sought applied as the universal solution, conceptual pitfalls occur if the understanding that agents are conceptualized as "multithreaded software" is not emphasized. Analysis and design pitfalls occur if the developer ignores related technologies. On the other hand, agent-level pitfalls may occur if the developer tries to use too much or too little artificial intelligence in the agent-system. Society-level pitfalls can occur if the developer sees agents everywhere or applies too few agents in the agent-system.

ENGINEERING MOBILE AND CONTEXT AWARE MULTIAGENT SYSTEMS

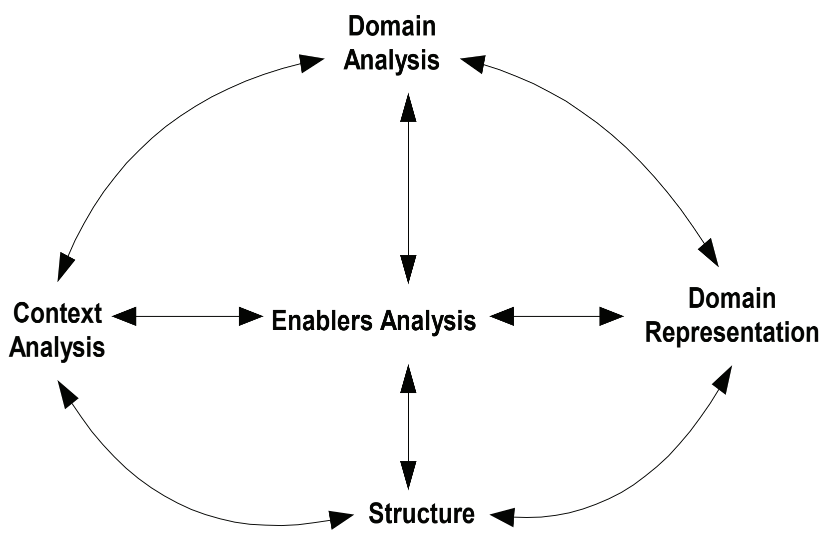
Despite the growing deployment of multiagent, mobile, and context-aware systems in different domains, little has been done with regards to the development of the methodologies that can be used to "engineer" them. While responses to complexities exhibited tend to be based on the reconfiguration of resources (hardware and software) and processes associated with tasks, models, databases, and interfaces (devices accessible for user input and display), current AOSE methodologies fall short to represent and manage the context of network processing capacity, organizational preparedness, connectivity, and cost. In addition, the proceduralization processes associated with multiagent, mobile, and context information increased the necessity of developing a consistent explanatory framework for explaining and anticipating the results of a decision or an action and therefore, deserves a paradigm shift with regards to software engineering methodologies and patterns. The proposed paradigm transformation needed (represented in Figure 1) should be reflected in the following aspects:

1. Agent modeling as well as data molding and management.
2. The spatial context and the use of integrated architectures such as service-oriented architectures.
3. Managing service push and agent pull matrix and mechanism.

Domain Analysis

Domain analysis aims at improving the understanding of problem domains and solution spaces by identifying decision-making partners and stakeholders together with their objectives, preferences, and utility matrix. Existing agent-oriented software engineering methodologies have done much emphasis on task decomposition, associating agent functionalities with assumed privileges and drawing up road maps for agent-oriented analysis of problem domains. However, despite the similarity of the tasks incorporated, the process of domain analysis is being approached differently by different methodologies. Some AOSE methodologies approach task decomposition and assignment of roles to agents by using the main object-oriented concepts in a way that allows for "encapsulation," "inheritance," and "reusability" of software modules and components. However, the process of coupling object-oriented concepts with agent-oriented ones has been challenged by the fundamental mismatch between the two concepts when they are implemented (Wooldridge & Jennings, 1999). To overcome this limitation, other AOSE methodologies approach domain analysis by incorporating agent-oriented concepts and theories of agent-hood into their models, knowledge base, and structures (Odell, Parunak, & Bauer, 2000; Wooldridge, Jennings, & Kinny, 2000). While each methodology is claiming comprehensive coverage of the phases of system development, the lack of maturity and inability to capture and represent the autonomous and proactive behavior of agents, as well as the richness of the interactions, have limited their deployment (Zambonelli, Jennings, & Wooldridge, 2001). Each successive development either claims to make the engineering process easier or promises to extend the complexity of applications that can feasibly be built (Jennings, 2001). While the main emphasis of all AOSE methodologies tend to be on drawing a framework for agent-oriented systems' analysis and design and developing implementation tools, they do not provide straightforward connections to the implementation of agent-based systems (DeLoach, Mark, Wood, & Clint, 2001; Wooldridge et al.,

Figure 1.



2000). The problem has been complicated by the emergence of the dimensions of context awareness and the expanding use of mobile multiagent systems for the implementation of mobile computing and other electronic transactions.

To take these variables into account, much emphasis needs to be made on domain analysis by crystallizing data access and using paths and processes associated with wireless mobile devices while they are operating in the active mode and the activities of their users when they are disconnected or using other devices. Such processes include the activities undertaken by mobile agents when they work to shift the workload from a low-capacity terminal to a server, or to filter or customize data before transfer. Incorporating these activities in domain analysis enables the development of efficient location mechanisms for mobile entities (Gasmelseid, in press). When emphasizing on decision-making environments, integrated domain analysis improves the understanding of the dynamics of using mobile agents and context-aware applications for dynamic optimization and control of decision support queries and their evaluation in a wide-area distributed domain of databases. Additional effort is needed to appropriately conceptualize a "context-aware" decision-making domain.

Context Analysis

The issue of context analysis relates to the creation of a context-rich decision-making environment that enables decision makers—while they are on move—to take decisions, monitor progress, and carry out tasks. The transformation of business enterprises and their migration to global complex operations has increased the need for real-time information and the use of intelligent Web-based "delegation" techniques on the one hand, and increased the importance of analyzing the "contextual" domains using Web technologies and the emerging data analysis and representation techniques (i.e., data mining, data warehousing, OLAP, etc.) on the other hand. While databases are currently developed as a part of the general system development process, contextual analysis enables the appropriate development and use of "mobile multidatabases," the articulation and refinement of the methods of information retrieval and decomposition of "intelligent" tasks. It also "reshapes" the rationale of domain analysis as it significantly affects the nature of information to be acquired, the methods to be used, and the way to be represented. Thorough context analysis also affects the nature of analytical tools to be used (including Web intelligence) and the management of information representation concepts (e.g., semantics, ontologies, and acquisition).

Because they used to be occupied with "task definition" and "role assignment," current AOSE methodologies do not pay enough attention to the "context of decision-making problems." The problem will be complicated by the introduction of context awareness and mobility considerations and the inability of many developers to associate "their development efforts" with the nature and context of decision making. When decision-making domains are characterized with the lack of relevant information, the process of context analysis will differ from the situation where decision makers are faced by an "over abundance" of irrelevant information. While emphasis of context analysis of the former situation tend to be oriented towards the acquisition of information from multiple and heterogeneous data sources to be used for the development of multiple databases and "indexation," the later situation demands emphasis on "data reduction techniques," "filtering," and information organization. Despite the growing use of context-aware systems, none of the existing AOSE methodologies are providing an in-depth context-based analysis.

Structure

In addition to their ability to represent linkages, "structures" integrate different architectural considerations governing the nature and magnitude of context awareness and mobility concerns to be experienced by agents either individually or collectively. While the structure dictates and significantly affects the "processing capacity" of the entire system, it also affects the structure and fabric of wired and wireless communication infrastructure as well as the necessary "file management" tasks and the automatic reconfiguration of network components. Especially in decision support systems, the incorporation of context awareness and mobility qualities also challenge the capacity of current AOSE methodologies to adequately capture an agent's flexible, autonomous problem-solving behavior, the richness of an agent's interactions, and the complexity of an agent system's organizational structures. Most of these methods feature a technology-driven, model-oriented, and sequential approach and assumes (in advance) the suitability of multiagent technology for the development of multiagent applications which may not always be the case in different problem domains. Because model orientations of these methodologies are obvious, the process of model coupling and integration process does not explicitly reflect the links between models. Besides the main issues (known as agent qualities) to be addressed by agent-oriented software engineering methodologies (such as autonomy, reactivity, proactiveness, and social ability), the concern for mobility have been growing over time (White, 1997). In spite of the growing diffusion of mobile agent technology, little research has been done to settle "design" directions to be followed in order to determine when mobile agents are convenient to be used or not. However, the current agent-oriented software engineering methodologies used for developing multiagent systems do not provide methods to determine in which cases mobile agents should be used. Many of the existing methodologies intentionally do not support intelligent agents; rather, they aim for generality and treat agents as black boxes (Padgham & Winikoff, 2002).

Domain Representation

Domain representation relates to the selection of software platforms, spatial and visual mechanisms necessary for the representation of contextual and noncontextual information as well as the technologies that make the acquisition, sharing, and use of such information possible. While different software platforms are being developed to support agent programming and context awareness, such efforts are still challenged by the lack of universally accepted programming languages and architectures that allow the incorporation of context awareness and mobility concerns in decision-making domains.

The process of information utilization in multiagent systems is challenged by the lack of communication languages geared towards direct machine-to-machine communication and the programming of hardware or software computer systems in accordance with a specific structure. Despite the fact that a number of agent communication languages have been proposed, notably hyper text mark up language (HTML), knowledge query and manipulation language (KQML), and Foundation for Intelligent Physical Agents (FIPA) agent communication language, so far, no standard exists to express the structure and content of messages and allow agents to understand each other. Furthermore, most of the languages do not specify syntax or semantics of the contents of the message leaving the relationship between logic and the interpreted programming language loosely defined.

Enablers' Analysis

Context awareness and mobility brings a wide range of new issues and/or increases the urgency and critical-ness of others. All concepts and issues that affect the four components shown in Figure 1 above are all called "enables." The set of enablers includes security considerations, change of user preferences and location, change of problem domains and solution spaces, authentication, and third party certification, efficiency of network, and connectivity and severity of database dependencies. Despite the variety of enablers and their situational-dependence, the focus of this chapter will be limited to the main two enablers that significantly shape software engineering efforts.

Authentication and Third Party Certification

The importance of "authentication" and third party certification stems from the interface matrix (among agents and between agents and their owners) as well as from the dynamics of "mobility and context awareness" associated with the mobility of agents to access different networks (for task handling and information acquisition), communicate with different service providers, and use multiple types of data and structures. Such considerations are associated with a wide range of agent-user (change of agent model base, knowledge engines, system infrastructure and architecture) and network-specific (change of infrastructure and architecture) considerations. While user-agent considerations reflect changes on "domains and context," network-specific ones bring load balancing and network optimization issues to the surface. Especially in decision-oriented, context-aware and mobile multiagent systems, the domain of decision making should not be left for full and absolute autonomy of agents. Because strategic decision making, for example, is oriented towards the external environment, the possibility of change is high, therefore. such consideration must be incorporated into the agent's development methodology.

On the other hand, the mobility and context-awareness of agents results in different modifications on the size of the agent and its ability (as a partner in a multiagent organization) to use load balancing mechanisms to manage context-based challenges. Throughout such movement and rooming the size of the entire agent may increase, the service provider misdirect it, other agents (absorb) it, and/or having other networks block its functionality as of change and frequent modifications of Web site addresses and/or data types. The entire agent-oriented software engineering methodology must incorporate suitable guidelines to be used for developing and mainstreaming of the necessary load balancing mechanisms and artifacts to be used by the entire agent or multiagent organization to destroy a part of the agent's contents (which part?), to pass it to the corporate knowledge base, or to limit its rooming, mobility, and context awareness by its capacity to exercise load balancing mechanisms. While these issues are affecting the efficiency of the entire multiagent organization, their importance looms very big when mobility and context-awareness issues are incorporated. However, despite the emphasis of current AOSE methodologies on "task decomposition" and "task-functionality" coupling, little has been done to approach theses issues.

Incorporating Security Mechanisms

The development of mobile and context-aware multiagent information systems increased the importance of information security by focusing on the use of technological solutions in terms of hardware devices or computer programs (e.g., cryptographic algorithms, digital signatures and challenge response authentication techniques, hash algorithms, and hybrid encryption mechanisms and protocols). The basic aim of such "technological" intervention mechanism is to prevent, avoid, detect, or prepare for breaches of security that threaten the confidentiality, integrity, or availability of information processed by computer systems.

While the entire agent-oriented software engineering methodologies have provided alternative ways for task decomposition and description of "relationships," little, also, has been done to incorporate "information security" considerations in multiagent "mobile" and "context-aware" applications (Gasmelseid, 2006). The importance of maintaining the information security of multiagent systems originates from two basic considerations:

a. The expanding use of these systems in different life-related domains such as banking, insurance, education, medicine, tourism, airlines, entertainment, and the management of pooled natural resources, among others; and
b. The growing number and type of threats associated with the enterprise transformations

and change of the qualities that constitute agent hood. Threats to multiagent systems range from uncontrolled accessibility and modification of core agent codes and services to the integrity of processes and communications, service execution, and coordination.

Incorporating information security considerations demands not only change of programs and technological solutions but also deserves a thorough "revisiting" of the way multiagent systems are being "engineering" and "developed." Accordingly, such paradigm shift necessitates increasing emphasis on the description and implementation of two basic issues of information security:

1. Linking information security with functional decomposition, message initiation, and communication in pursuit of integrated multiagent task management; and
2. Viewing security mechanisms on "layered bases" in order to enhance information flow, mainstream feed back, and use appropriate metrics and standard operating procedures to reduce threats at earlier stages.

The decision-making context reflects the entire organizational structure (through which information is exchanged), objective (to be supported by the multiagent information system), and the decision-making models (reflecting managerial styles). The importance of maintaining task-security coherence and layered information security is affected by the increase and diversity of the devices and resources to be used in heterogeneous networks and the functions to be performed in spontaneous ad hoc communications in a transparent, integrated, and extensible fashion.

Approaching information security through technological solutions is challenged by the variety of key length, computational complexity, and breach possibilities. Therefore, it is essential to "couple" technological solutions with an array of other factors (i.e., human resources, standard operating procedures, structure, and system development methodologies) that should be investigated when addressing information security. The importance of analyzing and understanding "enablers" originates from the fact that multiagent, mobile, and context-aware systems use multidatabases and multi-DBMSs that may present some obstacles generated by local database heterogeneity. While the summary schemas model can be used in multidatabases as a solution that utilizes hierarchical metadata in which a parent node maintains an abstract form of its children's data semantics in intelligent systems, the analysis of enablers looms very big. For the hierarchical structure and the automated schema abstraction of the entire multiagent, mobile, and context-aware system to significantly improve the robustness and generate a dynamic expansion capability, the analysis of enablers facilitate the incorporation of security and integration dimensions in the body of software engineering methodologies and allows for early warning and prompt response.

However, revisiting AOSE methodologies to incorporate information security considerations has three reflections on the functionality of multiagent, mobile, and context-aware systems. (a) It "embodies" security in solution spaces when specifying and structuring problems in accordance with rules, behavior expectations, and authority relations particularly in open dynamic environments. (b) The emphasis on AOSE-based information security measures, rather than technological solutions only, brings "ontological and semantic" considerations to the surface at early stages. (c) Also, it significantly affects the trade off among alternative agent architectures.

FUTURE TRENDS

The migration towards Web-based, context-aware, and mobile systems warrants more understanding of the way such AOSE methodologies are used. The use of ad hoc networks and mobile components "databases and devices" to support the mobility of users incorporates some network-specific, user-oriented, and third party-based functionalities and processes that should be reflected by AOSE methodologies in use together with the appropriate metrics, standard operating procedures, and feedback mechanisms that comply with “mobility” and “contextual” requirements.

The projection of future trends shows more deployment of mobile commerce applications of the dynamics of the international economic system, organizational transformations, and changes of life styles. Within this context, more emphasis will be made on integrating features from different computing paradigms (i.e., mobile, pervasive, ubiquitous, grid, and soft) as an attempt to relax “architectural” limitations. To streamline functionalities, more focus is expected to be made on the orchestration of mobile and context-aware applications by maximizing benefits of Web technologies and managing “ontological and semantic” problems.

On the other hand, future trends can also be seen in organizational landscapes where organizations will continue to invest in improving their learning curves and turning into “learning organizations” by focusing on context awareness and mobility and maintaining linkages with decision-making domains. Future trends also show that additional R&D efforts will be directed towards relaxing the difficulties associated with context acquisition, representation, and sharing.

CONCLUSION

The focus on “responsiveness” and “coordination” in today’s digital economy is being motivated by the trends of competition, market fragmentation, and shift of consumer preferences. To ensure “safe migration” from “conventional” to “electronic” business domains, enterprises have been increasingly concerned with “revisiting” their decision-making styles and decision support systems in pursuit of achieving objectives. The unprecedented technological developments (i.e., hardware, software, and telecommunications) exhibited over the last couple of years has been accompanied by a wide range of concepts (such as context-aware systems, ubiquitous computing, wearable computing, and pervasive computing) which have significantly reflected “some fragmentation” of decision-making processes across an archipelago of geographical locations and increased the interest of decision makers in “context-related” information as a complementary for internal and external organizational repositories of information to support mobile applications.

Despite the expanding use of mobile applications and the emergence of context-aware ones little has been done to “reinvestigate” the potential of current AOSE methodologies and their potential to represent mobility and context awareness considerations. While some focus has been made on “domain analysis,” additional efforts are needed with regards to “context analysis,” “domain representation,” “structures,” and linking them with the set of enablers in an integrated way.

REFERENCES

Booch, G. (1994). *Object-oriented analysis and design* (2nd ed.). Reading, MA: Addison-Wesley.

Cabri, G., Leonardi, L., & Zambonelli, F. (2000). Mobile agent coordination models for Internet applications. *IEEE Computer*, *33*(2), 82-89.

Chatzipapadopoulos, F., Perdikeas, M., & Venieris, L. (2000). Mobile agent and CORBA technologies in the broadband intelligent network. *IEEE Communication Magazine*, 38(6), 116-124.

DeLoach, S., Mark, A., Wood, F., & Clint, H. (2001). Sparkman. Multiagent systems engineering. *International Journal of Software Engineering and Knowledge Engineering*, *11*(3), 231-258.

Du, T., Li, E., & Chang, A. (2003). Mobile agents in distributed network management. *Communications of the ACM*, *46*(7), 127-132.

Ee-Peng, L., & Keng, S. (2001). Mobile commerce. *Journal of Database Management*, *12*(3), 1-3.

Erol, K., Lang, J., & Levy, R. (2000). Designing agents from reusable components. In *Proc. of the Fourth Int. Conf. on Autonomous Agents* (pp. 76-77).

Gasmelseid, T. M. (in press). Engineering multiagent systems. *Encyclopedia of information security and ethics*. Hershey, PA: IGI Global.

Gasmelseid, T. M. (2006a, April 22-24). A multi agent negotiation framework in resource bounded environments. In *Proceedings of the ICTTA06*, Damascus Syria.

Gasmelseid, T. M. (2006b). Multiagent Web based decision support systems for global enterprises: An architectural blueprint. *Engineering Letters*, *13*(2), 173-184. Hershey, PA: IGI Global.

Gasmelseid, T. M. (2007, July-September). A multiagent service-oriented modelling of e-government initiatives. *International Journal of Electronic Government Research*, *3*(3), 87-105.

Gerhard, W. (2002, January). Agent orientation in software engineering. *Knowledge Engineering Review*, *16*(4), 349-373.

Huang, M., John, T. Robert, S., & Towers, M. (1995). *Understanding the context of Medical Informatics*. Tokyo: Robertson Press.

Jain, A., Aparicio, M., & Singh, M. (1999). Agents for process coherence in virtual enterprises. *Communications of the ACM*, *42*(3), 62-69.

Jennings, N. (2001). An agent-based approach for building complex software systems. *Communications of the ACM*, *44*(4), 35-41.

Jennings, N. R., & Wooldridge, M. (1998). *Applications of intelligent agents*. Retrieved September 18, 2007, from http://agents.umbc.edu/introduction/jennings98.pdf

Jennings, N., & Wooldridge, M. (1999). Software engineering with agents: Pitfalls and pratfalls. *IEEE Internet Computing*, *3*(3), 20-27.

Jennings, N., & Wooldridge, M. (2002). Agent-oriented software engineering. In J. Bradshaw (Ed.), *Handbook of agent technology*. AAAI/MIT Press.

Keng, S., Ee-Peng, L., & *Zixing, S.* (2001). Mobile commerce: Promises, challenges, and research agenda. *Journal of Database Management*, *12*(3), 4-10.

Odell, J., Parunak, H. V., & Bauer, B. (2000). Extending UML for agents. In G. Wagner, Y. Lesperance, & E. Yu (Eds.), *Proceedings of the Agent-Oriented Information Systems Workshop at the 17th National conference on Artificial Intelligence,* Austin, TX (pp. 3-17).

Padgham, L., & Michael, W., (2002). *Prometheus: A methodology for developing intelligent agents.* Retrieved May 20, 2007, from http://www.cs.rmit.edu.au

Parunak, H. (1999). Industrial and practical applications of distributed AI. In G. Weiss, (Ed.), *Multi-agent systems* (pp. 377-421). MIT Press.

Sandholm, T., & Huai, Q. (2000). Nomad: Mobile agent system for an Internet-based auction house. *IEEE Internet Computing*, *4*(2), 80-86.

Satyanarayanan, M. (2001). Pervasive computing: Vision and challenges. *IEEE Personal Communications*, *8*(4), 10-17.

Wei, G. (2001). Agentenorientiertes software engineering. *Informatik Spektrum*, *24*(2), 98-101.

White, J. (1997). Mobile agents. In A. Bradshaw (Ed.), *Software agents* (pp. 437-472). AAAI Press.

Wooldridge, M., & Jennings, N. (1995). Agent theories, architectures and languages: A survey. *Lecture Notes in Computer Science*, *890*, 1-39.

Wooldridge, M., Jennings, N., & Kinny, D. (2000). *The Gaia methodology for agent-oriented analysis and design*. Kluwer Academic Press.

Yu, J., & Hurson, A. R. (2004, October-December). Application of mobile agents in mobile data access systems: A prototype. *Journal of Database Management*, *15*(4), 1-24.

Zambonelli, F., Jennings, N., & Wooldridge, M. (2001). Organizational rules as an abstraction for the analysis and design of multi agent systems. *International Journal of Software Engineering and Knowledge Engineering*, *11*(3), 303-308.

KEY TERMS

Agent-Oriented Software Engineering: It is the process or methodology of developing and engineering software that has the concept of agents as its core computational abstraction, particularly in the context of complex, open, networked, large, and heterogeneous applications. The main purposes of agent-oriented software engineering (AOSE) are to create methodologies and tools that enable inexpensive development and maintenance of agent-based software in a flexible, easy-to-use, scalable, and quality fashion.

Context Awareness: It is the feature that reflects the ability of mobile devices to acquire and use information about the circumstances under which they operate and can react accordingly. By making assumptions about the user's current situation, context awareness assists in the design of user interfaces in ubiquitous and wearable computing environments and hybrid search engines. It reflects any task-relevant information and/or service that can be used to characterize the situation of an entity to a user.

Context Management: Denotes all activities related to the acquisition of context-related information and developing integrated frameworks that maintain linkages between the domain of decision making, architectural styles, and technological components.

Decision-Making Context: A decision-making context can refer to the "situation in which decision-making processes take place among different decision partners (denoted as objects including customers, suppliers, and governments)" with special emphasis on the change agents incorporated in the decision-making environment (i.e., those influenced by and influencing decision-making processes such as mediators and competitors) irrespective of the location at which decisions are taken.

Mobility: As a key quality of multiagent and context-aware systems, mobility is concerned with providing universal access to communication tools, networks, databases, and information repositories, as well as reliable applications, regardless of their location or type of access devices. It can be implemented through calling, conferencing management, presence, messaging management, contact and information management, and personal efficiency management and other tools that guarantee access to messages represented in different data formats.

Multiagent Systems: A multiagent system is a collection of, possibly heterogeneous, computational entities that use their own problem-solving capabilities to interact in order to reach an overall goal. Their ability to improve information availability, problem solving capabilities, corporate control, and distributed data processing gave them more importance in different domains of application. They proved to be suitable for complex, distributed problems involving a multiplicity of interconnected processes whose solutions demand the allocation of fusion of information and expertise from demographically distributed sources.

Multidatabase Systems: They represent the global (federated and nonfederated) systems on top of the existing heterogeneous local databases that generate an impression of uniform access with reasonable cost. Nonfederated database systems do not support local autonomy but federated database systems do. A federated databases system consists of component databases that are autonomous and sharable. To overcome the local schema heterogeneity problem and support global transactions, federated database systems normally adopt the layered schema architecture to orchestrate heterogeneous local-level data models with the uniform global ones (canonical or common data models). When using multidatabase systems, special attention should be paid to the management of schema redundancy that may exist between different layers, and the maintenance and manipulation of the global-level schema.