Decision Making in Distributed Objects Using Radio Frequency Identification

Kathleen S. Hartzel, Palumbo-Donahue School of Business, Duquesne University, USA
William E. Spangler, Palumbo-Donahue School of Business, Duquesne University, USA
Mordechai Gal-Or, Palumbo-Donahue School of Business, Duquesne University, USA
Jerrold H. May, J. M. Katz Graduate School of Business, University of Pittsburgh, USA

ABSTRACT

This paper explores the architectural and management challenges related to the structure of a distributed production control application and its integration with the physical hardware comprising the production line. We propose that management control, risk management and information management can be enhanced through an integrated, agent-based systems architecture that attaches data processing and storage capabilities to physical objects. The objective of our research is to extend prior research and development in integrated systems architectures by using radio frequency identification (RFID) technology. RFID technology will allow much of the coordination and decision making to take place directly in the RFID tags, rather than in digital surrogates within the software application. This projects the digital representation directly into the physical world, minimizing the intermediary and somewhat centralized software-based system, and creating a more natural representation of a physical, object-based system.

INTRODUCTION AND MOTIVATION

Advances in sophisticated data storage and wireless communications technologies, such as radio frequency identification (RFID) devices, provide opportunities to incorporate object-oriented (O-O) processing and design techniques within the physical world. (Nicolai et al., 2005) This is a natural evolution of the object-oriented software paradigm, given that intelligent software objects, or agents, are intended to mimic the structure and behavior of entities in the real world, thereby allowing a more 'natural' model of a complex engineering, biological or social system. (Jennings, 2001) Likewise, the application of O-O design and control techniques within an RFID-enabled system can lead to improved flexibility, agility and robustness in a variety of physical environments, including transportation, health care, and inventory control. (Nath et al., 2006)

This paper proposes that management control, risk management and decision making within a production environment can be enhanced through an integrated systems architecture that incorporates decision making capabilities within physical objects. We focus specifically on the design and control issues of an object-oriented distributed application involving physical objects, such as machines and materials, casting them as intelligent agents in the context of a potential implementation environment such as RFID. The O-O implementation intentionally leads to a distributed, loosely coupled information system infrastructure, where processing, data storage, and control is assigned, via RFID tags and technology, to physical objects making decisions within the production domain. Assigning distributed responsibilities to individual objects raises important questions about the ability of individual RFID tags to act contingently in a dynamic physical environment. That is, to what extent can a tag plan or re-plan based on environmental parameters such as its location, date/time, and proximity to other tags or readers? Furthermore, how is this contingency planning controlled by the overall system?

In many respects, such control issues are addressed within the context of an object-oriented design. An object- or agent-oriented approach is motivated by a disconnect between the physical operation of a system, which is inherently decentralized, and its management and control structure, which tends to be centralized. Consider, for example, a manufacturing environment in which the production line is dynamic and produces a number of different items. The actual execution of the system – including operations, queues, workstation setups, and movement of items – is physically decentralized. By contrast, the associated manufacturing and control system is managed through a centralized control mechanism, which produces the master production schedule, work orders, change orders, and expedite orders, as well as all status and performance reports. Arguably, the complexity of a physically distributed system makes a centralized control strategy inherently less efficient and more prone to error.

The complexity of the example manufacturing environment makes it well-suited to a decentralized, agent-based implementation strategy in which the physical components themselves interact with one another as part of a coordinated assembly and delivery system. (Jennings, 2001) Because objects inherently exhibit independent behavior and private information, they are capable of acting flexibly and autonomously within a decentralized environment. In the example environment, each of the components is capable of storing knowledge of its individual state and objectives over time, and then communicating that knowledge to other components throughout the duration of the manufacturing process.

Creating a decentralized control environment requires a technology that can implement data storage, processing and communication capabilities of software objects within objects in a physical environment. Although RFID faces a number of engineering-related challenges, including environmental interference and the cost of implementing and using individual tags (Want, 2006), successful implementation of an object-oriented, real-world system also requires a management model that guides the design and implementation of the system. The control structure of a distributed, RFID-based system must provide an alignment between the overall objectives of the business and the functionality of the implemented system, which is a self-organizing and potentially self-interested set of independent physical objects. From a governance perspective, those objects should operate cooperatively in pursuit of shared organizational goals, and they should do so in a manner that adds value beyond the traditional centralized approaches to systems design.

A PRODUCTION LINE EXAMPLE

We illustrate these issues using an production example based on Bussman and Schild, wherein an industrial consortium led by DaimlerChrysler developed an agent-based flexible production system. (Bussmann and Schild, 2001) Figure 1 shows a simplified version of the production line. The line consists of a set of machines (designated as M_i) connected by a conveyer belt, with different items (represented as circles) transported via the conveyer belt for processing by specified machines. Each machine is capable of performing a certain set of operations (designated as O_i). As the production process unfolds, items proceed in a circular fashion around the production floor. If a machine has the capacity and availability to serve an item moving on the conveyer, the conveyer directs the item to that machine. Upon completion of the operation, the item returns to the conveyer and awaits the next machine capable of processing its next operation.

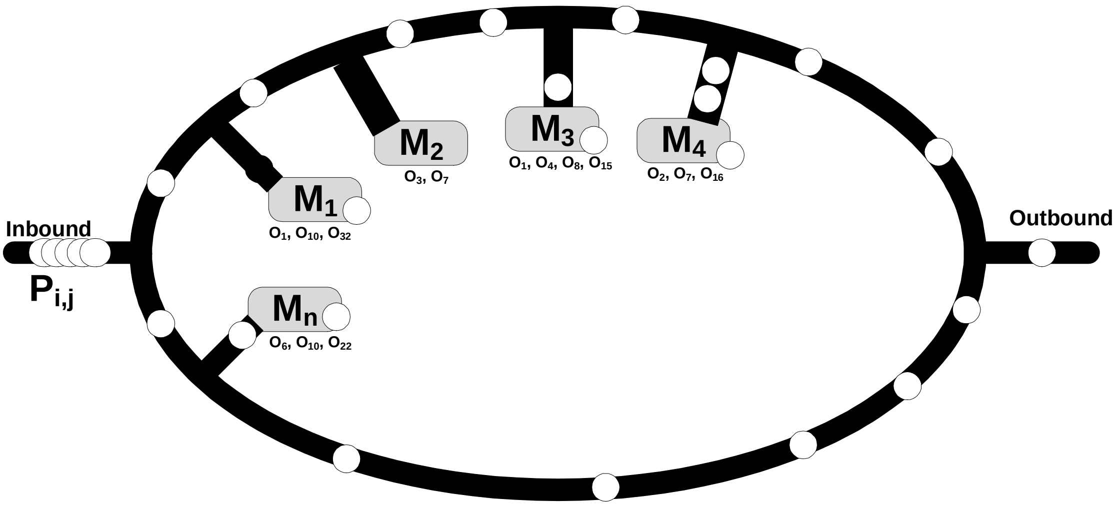

Figure 1. An agent-based production line composed of machines (M_1-M_n), items, and a conveyer belt (adapted from Bussmann and Schild, 2001)

Despite the implied simplicity in this example, the physical operation and control of the system is quite complex. First, the time required to process an individual item varies from machine to machine and from item to item, and is dependent on a number of random elements inherent in the system. That randomness stems from the type of item being served as well as the set of tasks a given machine has performed prior to serving the item. Furthermore, the performance of any given machine cannot be predicted with certainty. A machine might deteriorate over time, thus requiring it to take longer to complete a certain operation. The machine also might become unavailable if its performance falls below a certain threshold, thereby requiring unscheduled maintenance, or if it breaks down and requires emergency repair.

The environment's inherent complexity argues for an alternative to an exclusively centralized control structure. The traditional, centralized approach used in computer integrated manufacturing requires anticipation of contingencies at design time, and considerable monitoring and re-planning by the central system throughout the production process. By contrast, Figure 2 illustrates the digital and physical representations of the agent-based approach taken by developers of the DaimlerChrysler production floor system. The system manages the complexity by delegating much of the decision making to the objects – i.e., the items and machines – on the production floor. In this system, each of the physical objects corresponds to an associated software agent in the computer application. The agent in turn acts as a decision-making proxy for its corresponding physical object, with the results of its decision communicated to other software agents within the application and, ultimately, back to the physical objects on the production floor. The physical objects on the production floor then act in accordance with those results.

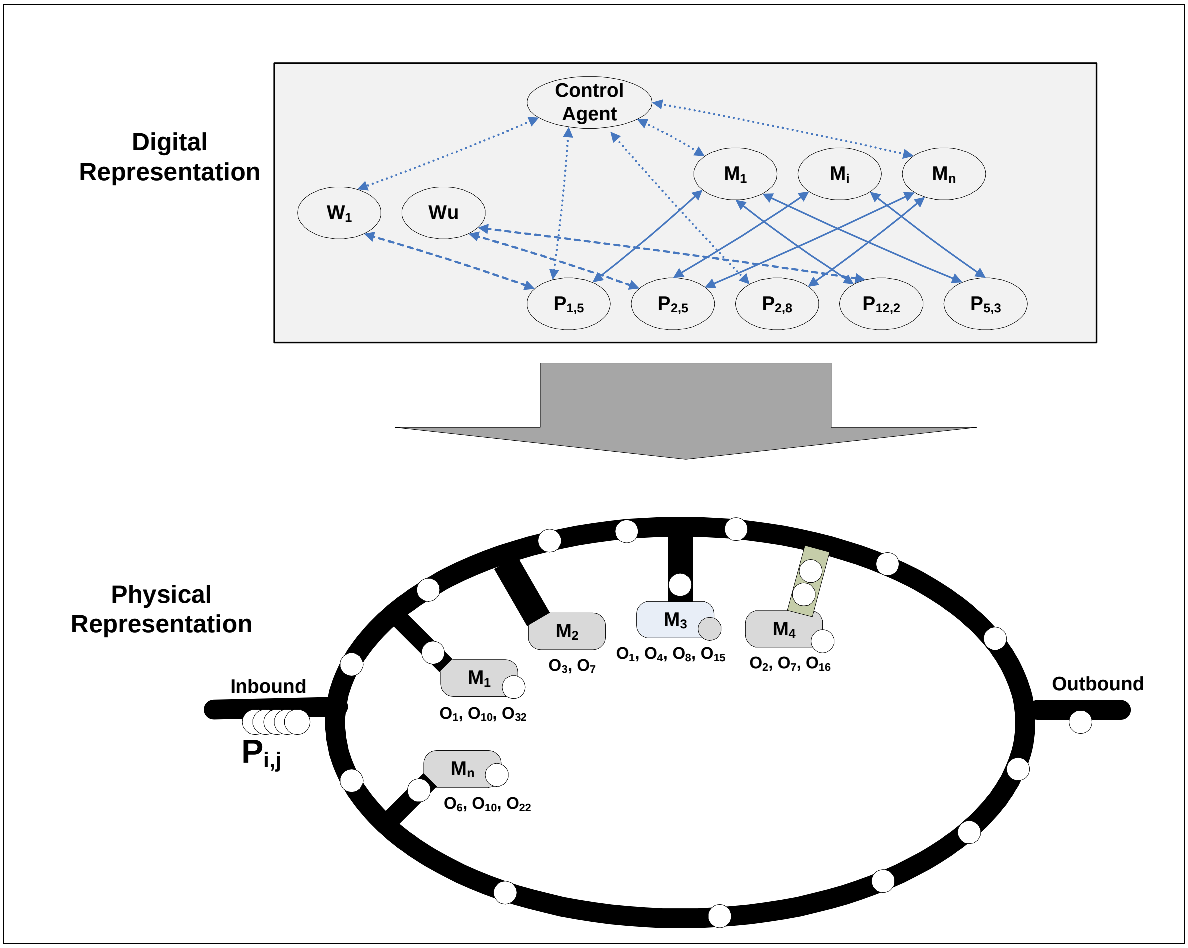

Figure 2. Modeling and controlling the production line using a digital representation, where Mi = machine, Oi = available operation, Pi = agent/object, and Wi = conveyer switch. Products-in-process are represented as red circles. (adapted from Bussmann and Schild, 2001)

The objective of our research is to extend this integrated systems architecture approach using RFID technology, which will allow much of the coordination and decision making to take place directly on the RFID tags. This in effect projects the digital representation into the physical world, which minimizes the intermediary and somewhat centralized software-based system, and creates a more natural representation of a physical, object-based system. We will return to that theme after a brief discussion of RFID technologies.

AN OVERVIEW OF RFID

At its most basic level, an RFID tag is a small device that can be attached to a product or container of products, and which can store and transmit a unique item or product identifier to readers at various locations over time. (Borriello, 2005) From an engineering perspective, the heart of an RFID tag is a microchip which, because of the need to minimize the cost of the tag, currently has limited processing capability and a finite amount of persistent memory. The tag also has an integrated antenna that is used to receive and transmit information, typically its identity. The tag's identity is persistent, resides in the tag's memory, and is communicated to readers through a standard interface.

Two characteristics of tags – i.e., the ability to transmit information via radio waves and to process and store information independently – distinguish RFID from other identifiers such as standard bar codes. First, because it does not need visual access between the tag and reader, the system is able to scan and track tags at very high-volume and in various positional configurations. This would allow the system, for example, to process an entire container of RFID-labeled products without necessarily unpacking the container. Second, whereas bar codes are capable only of recording and communicating a static identifier, an RFID tag is able to receive information, process it in an intelligent manner, and output the results of its processing to a diverse set of readers and other tags. Currently, RFID technology ranges from *passive* tags, which require an external source of power and a reader to initiate data transfer, to *fully active* tags, which feature their own power source as well as greater processing power and data storage capabilities. As such, active tags are capable of more sophisticated and independent behaviors – such as initiating peer-to-peer communication with other devices and acting based on information received. Over time, as RFID technical capabilities improve, tags should become increasingly capable of sophisticated environmental monitoring and decision-making, and thus begin to act as intelligent agents on behalf of their associated real-world objects. In the manufacturing example above, for example, advanced RFID tags would allow sophisticated processing and communication between items and machines, allowing items to self-direct from machine to machine based on local knowledge of their individual processing requirements.

APPLYING AN OBJECT-ORIENTED PARADIGM TO RFID DESIGN

For the reasons cited above, we propose an object-oriented approach to the production problem outlined in Bussman and Schild (Bussmann and Schild, 2001), based on the use of RFID technology to implement localized decision making by distributed agents in the production line. By using tags and readers attached to machines, conveyer belts, and items, we transform the physical entities into physical agents that direct the production process. As shown in Figure 3, readers would be attached to the machines and conveyer belts, and tags would be attached to items served. Each machine would be aware of the operations it could perform and its availability, and each item would be aware of the operations it requires and the ongoing state of the execution of those operations.

An object-oriented approach inherently supports the design, implementation and control of this type of distributed RFID system. An O-O system is not constrained by a single, pre-defined control structure that must anticipate every environmental contingency. Instead, many decisions can be pushed down to the local level and handled by the individual objects and/or their digital proxies. This results in a more flexible and robust system that is inherently capable of dynamically responding to the environment.

The defining characteristics of object-oriented systems also are compatible with distributed control of an RFID system. At the component level, each individual object has the following characteristics:

Identity: an object has a unique label or identity. The O-O notion of identity is analogous to an RFID tag's electronic identifier, which is a unique binary number that identifies the tag, and by extension the specific real-world object associated with the tag.

Properties: an object contains internal data in the form of attributes, which can be either static or dynamic. Static data would identify certain fundamental characteristics of the object, such as its creation date, description, operations required during its manufacture (as in the earlier example) and so on. Dynamic data would reflect changes to the object over time and/or logs of its processing history, such as whether or not it is currently queued for a machine, operations performed on the item and the machines that performed those operations, as well as relevant environmental readings such as temperature and humidity.

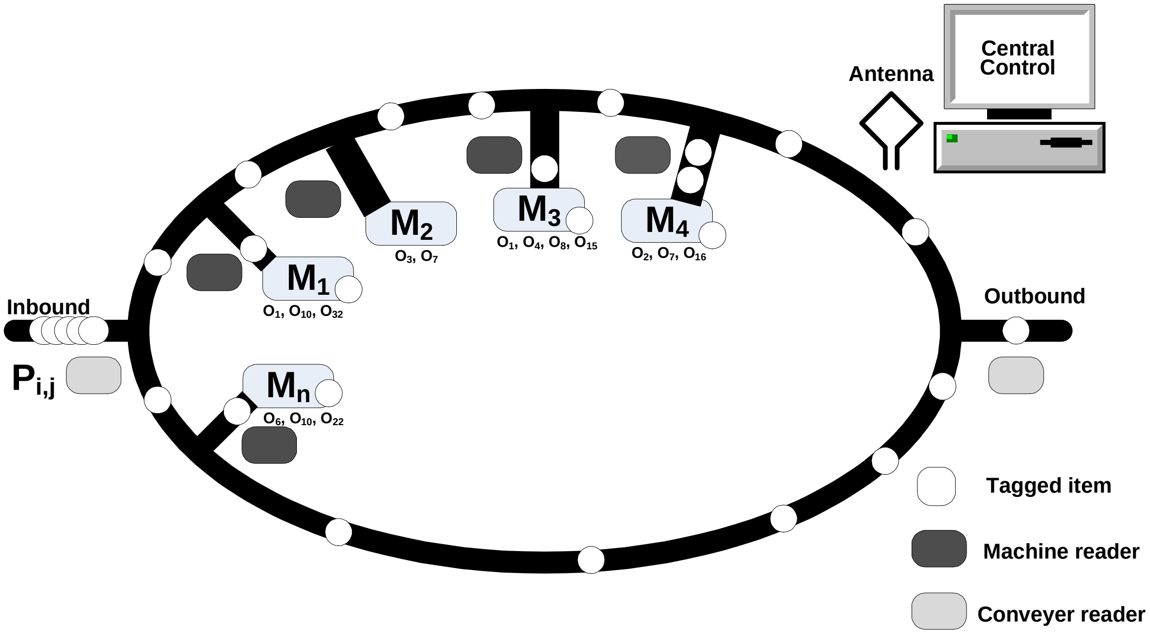

Figure 3. An agent/object-based production system implemented via RFID readers and tags.

Behavior: an object's behavior is defined by the methods or operations contained within the object. A method represents the procedural knowledge of an object. It acts on the object's data, and is invoked through procedure calls to the method. Similarly, because it is a microprocessor, an RFID tag also is capable of storing and executing embedded code. Its ability to send and receive data via wireless transmission means that a tag can communicate with other tags through procedure invocation. In our manufacturing example, an item would exhibit behavior by requesting a specific operation from an available machine, or by dynamically adjusting its priority when the wait time for a machine exceeds a particular threshold.

State: an object's state is determined by its properties and constrains its behavior. For example, in order to be placed in a machine's queue for a particular operation, an item's *state* should be 'all prerequisite operations completed' (or something similar). Any other state would prohibit the item from requesting that particular operation.

Inherent in an object-oriented approach are additional defining characteristics that are transferable to a physical RFID application. They include:

Encapsulation

In a traditional object-oriented system, an object can hide the details of its data and methods, allowing access only through its publicly-available methods, or *interface*. From a systems control perspective, this provides two benefits: 1) it provides an access control mechanism to protect internal and potentially sensitive information inside of the object, and 2) it relieves the calling object or client of any need to understand the details of the called procedure. This second benefit addresses the complexity issue by delegating specialized responsibilities to specific objects, which carry out those responsibilities without imposing the operational details on the centralized control system.

As a physical-world implementation of an encapsulated object, an RFID tag can behave similarly. The procedural code within a tag can be designed to respond to wireless interrogations on a selective basis, providing access to its internal data only through public methods. Those methods may require additional levels of security, including the use of passwords and encryption schemes. From a complexity perspective, information requestors such as other tags and readers do not have to know how the tag responds to and processes information. The requestors simply need to know which methods are available to call, which arguments must be passed to each method, and what information will be returned as a result. For example, in requesting service from an available production machine, an item needs to know whether a machine is available to perform the operation, and it perhaps needs some indication of the waiting time. But it does not need to know the procedural details regarding how a particular machine carries out the operation.

Generalizability and Inheritance

A powerful feature of object-oriented systems is their ability to define a hierarchical taxonomy of objects, in which individual objects share certain basic characteristics, including data attributes and methods. For example, while the various types of items in the production process are different in many ways, they nevertheless share common attributes such as description, creation date, and required operations. Furthermore, each item object has methods for communicating its description, adjusting its priority, and so on. In other words, each object *inherits* a set of common attributes and methods from the superordinate class. While the details regarding the objects' internal data and processing will vary, any calling object knows that it can send a particular message to *any* object of type *Item*, and that it can expect that object to contain and communicate particular types of data.

From a change management perspective, generalizability simplifies the model of the physical world and allows designers to respond in a more robust manner to required changes. Processes that are shared by the class or sub-class of objects (tags) can be changed in a single location at the class level, and then inherited by each of the member objects. Likewise, new data elements and communication protocols also can be defined for the entire class. The result is a reduction in design complexity and a corresponding reduction in the potential for design and/or programming errors.

Polymorphism

A side-benefit of encapsulation and generalizability is that the 'need to know' aspect of message passing can be made contextual, meaning that identical messages passed to different individual objects will produce conceptually equivalent responses, even though the responses were produced using significantly different internal processes. This capability is called *polymorphism*. For example, an item on the production line could pass the same 'request operation (O_1)" message to three individual production machines. The item neither knows nor cares how or whether the operation is performed by each machine, nor that the operation might be performed differently by the different machines. But the item does know that it can pass the same message to any appropriate object – e.g., one capable of performing an operation – and receive the same type of response from each. In this case, the item simply requests service by passing a message to the machines, and receives a response indicating availability and anticipated wait time (perhaps measured by the length of the machine queue). Again, this is a powerful technique for managing and controlling complexity, since each object-item does not have to tailor its communication protocol for each individual object-machine.

In an object-oriented RFID application, polymorphism would provide similar benefits by allowing tags to respond *appropriately* (and only) to *known* interrogations from other real-world objects. For example, in a production control environment, readers and other objects can send a single message to myriad tags asking for standard information such as transaction and location histories, associated components, weights, contents, and so on. Again, this reduces the complexity of process control, and it provides a layer of access control over the information contained in the tags.

MANAGEMENT AND CONTROL ISSUES

From a systems architectural and control perspective, the primary issues facing IT managers come from three areas: management control, information management, and risk management. We briefly summarize those areas below.

Management control

The key objective of any business application is to achieve a defined set of business goals. From the organization's perspective, goal achievement requires a mechanism by which the numerous local decisions made by independent objects can be coordinated in order to achieve global business objectives. In software-based O-O systems, process control typically is moderated, in a loosely-centralized sense, through a centralized component which manages the overall sequence of operations. The control object provides an agenda and a set of milestones for the system and its component objects to accomplish. Messages passed to each individual object prompt the object to act toward the achievement of its defined set of sub-goals.

Important questions concern the ability of individual RFID tags to make independent but coordinated decisions in a dynamic physical environment. That is, to what extent could a tag plan or re-plan based on environmental parameters such as its location, date/time, and proximity to other tags or readers? Furthermore, how is contingency planning controlled by the centralized component? Various architectural schemes have been proposed for controlling complex, distributed systems, many of which involve the use of *semi*-autonomous objects. These include middleware solutions, wherein message flow between objects is directed and filtered by an additional layer of software (Lupu, 1999) (Chawathe et al., 2004).

Information management

An integrated systems architecture, where processing and data storage capabilities are attached to physical objects, necessarily complicates information management. But this integrated architecture also creates new opportunities in processing, securing and maintaining information. Diekmann et al describe an RFID environment that encompasses two basic implementation schemes.(Diekmann et al., 2007) The more traditional and common approach is termed *data-on-network*, in which data are stored centrally on the network and then referenced by individual EPCs. The newer and less common approach is termed *data-on-tag*, in which data resides in traditional centralized repositories (i.e., databases) as well as within individual objects. Diekmann et al note that the RFID community's historical bias toward the data-on-network approach is driven primarily by engineering and cost issues, and it is clear that process and data management of the type discussed here would inevitably increase the required functionality of the tag, and by extension, its cost. However, with expected improvements over time in RFID functionality and decreases in cost, data management is not necessarily captive to physical constraints over the medium- to long-term.

Diekmann et al suggest that the data-on-tag concept should be considered for a number of reasons, most of which are consistent with arguments for a traditional object-oriented approach: 1) storing data on the tag is a complementary rather than competing strategy, in that a combination of centralized and distributed data storage makes the system more robust and less susceptible to data loss, 2) data can be collected at the time events in the real-world occur, thereby making data retrieval and analysis more immediate, 3) individual objects (tags) have access to the data they need to make decisions, and 4) much of the processing load is off-loaded from the central systems to the individual tags. (Diekmann et al., 2007) At the same time, a number of practical issues related to the data-on-tag approach must be addressed, including limited tag storage capacity, the high cost of technically sophisticated tags, and the availability of individual objects via the network at the time data are required. (Diekmann et al., 2007) Again, however, these tend to be engineering and production issues, and as such are relatively transient.

Risk management

While management control is a critical component of an overall risk management strategy, a significant number of other issues must be addressed at the implementation level. In this regard, even a 'traditional' object-oriented system raises a number of control and risk issues, and as such those issues also apply to an object-oriented RFID application. For example, because O-O is (still) an unfamiliar paradigm for many analysts, managers, auditors and software developers, the tools and techniques used in O-O design and programming also remain somewhat unfamiliar to many, which can lead to poorly-designed and poorly-documented application code (Cobb et al., 2007). A poorly-designed system might function inefficiently or inaccurately, or at the very least it might not take full advantage of RFID capabilities.

Security and privacy are important risk management issues in both object-oriented and RFID systems. From an RFID perspective, the fear is that data resident on an RFID tag can be easily copied or stolen, which is a violation of security (from the company's perspective), as well as a violation of privacy (from a consumer's perspective). While many of the proposed technical remedies are beyond the scope of this discussion, it is important to note that an object-oriented approach, and the control capabilities discussed above, potentially can address many of the concerns. A key to security in a distributed object-oriented system is the implementation of access control at various levels. By providing the RFID tag with sufficient intelligence to make control decisions in a complex, dynamic environment, an object-oriented approach facilitates the use of an in-depth, multiple-level approach to RFID security.

In a more general sense, O-O systems have various control features that aid in managing risk. Cobb et al have described a number of control features of O-O systems, including 1) the ability of many O-O computer-assisted software engineering (CASE) tools to automatically generate system documentation when the application code is developed, 2) the use of standard object-oriented (UML) diagrams to indicate permissible interactions between objects, 3) the definition of controls at the class level, with those controls then pushed down to the individual objects via inheritance (thereby ensuring that required controls are propagated appropriately across the system), and 4) the use of encapsulation and polymorphism to help ensure that data inside of an object are specifically under the control of that object, and that the access to the data is restricted on a 'need to know' basis. (Cobb et al., 2007)

CONCLUSION

Although researchers continue to advance the integration of digital and physical environments through RFID and related technologies, the pace of advancement has been somewhat slow. (Bell and Dourish, 2006) Because the current state of the art is constrained by engineering limitations on the data and processing capabilities of RFID tags and sensors, research efforts tend to focus on proof-of-concept applications dealing with everyday objects, such as automated recipe generators (Langheinrich et al., 2000), quasi-intelligent cell phones (Loke, 2006), furniture (Microsoft, 2007), and so on. Many of the applications are comprised of 'context-aware artifacts', or physical objects, which are able to reason in a simple manner based on contextual elements such as location, orientation, motion, temperature, and the presence of other known objects.

Loke notes that current context-aware applications are characterized by two approaches: 'self-supported', in which essentially all of the intelligence and control is contained within the individual objects, and 'infrastructure-based', in which a central controlling entity establishes the context and provides an integrating theme that is beyond the capabilities of the individual objects. (Loke, 2006) In automated recipe generator described in (Langheinrich et al., 2000), for example, the infrastructure monitors the addition of various tagged ingredients into the context, and proposes recipes based on the current collection of ingredients – something that individual ingredient-objects are incapable of doing. As noted above, Diekmann presents a similar dichotomy with regard to data-on-tag versus data-on-network.(Diekmann et al., 2007) The data-on-network approach, in which the RFID tag simply provides a pointer to a centrally-stored data record, most closely typifies the current state of RFID. In both the infrastructure-based and data-on-network approaches, a more centralized control structure is required, in part, to compensate for current limitations in processing and data storage at the tag level.

Nevertheless, as the technical capabilities of RFID and related hardware improve, the sophistication of potential applications are expected to improve as well. This is likely to lead to increased research and development of applications based on the more distributed approaches – i.e., 'self-supported' and 'data-on-tag'. As noted, with more objects making increasingly sophisticated, independent decisions, the level of system complexity will rise significantly. With that rise in complexity, management's focus necessarily will turn to a system design strategy capable of controlling it.

REFERENCES

Bell, G. & Dourish, P. (2006) "Yesterday's tomorrows: notes on ubiquitous computing's dominant vision " *Personal and Ubiquitous Computing,* 11**,** 133-143.

Borriello, G. (2005) "RFID: Tagging the World." *Communications of the ACM,* 48**,** 34-37.

Bussmann, S. & Schild, K. (2001) "An Agent-based Approach to the Control of Flexible Production Systems." *Proceedings of the 8th IEEE International Conference on Emerging Technologies and Factory Automation.*

Chawathe, S. S., Krishnamurthy, V., Ramachandran, S. S. & Sarma, S. (2004) "Managing RFID Data." *Proceedings of the 30th VLDB Conference.* Toronto.

Cobb, A. T., Guan, J. & Levitan, A. S. (2007) "Control Considerations in Object-oriented Systems." *Information Systems Control Journal,* 3**,** 28-33.

Diekmann, T., Melski, A. & Schumann, M. (2007) "Data-on-Network vs. Data-on-Tag: Managing Data in Complex RFID Environments." *Proceedings of the 40th Hawaii International Conference on System Sciences.*

Jennings, N. R. (2001) "An Agent-Based Approach For Building Complex Software Systems." *Communications of the ACM,* 44**,** 35-41.

Langheinrich, M., Mattern, F., Römer, K. & Vogt, H. (2000) "First Steps Towards an Event-Based Infrastructure for Smart Things." *Proceedings of the Ubiquitous Computing Workshop.*

Loke, S. W. (2006) "Context-Aware Artifacts: Two Development Approaches." *IEEE Pervasive Computing,* 5**,** 48-53.

Lupu, E. C. (1999) "Conflicts in Policy-Based Distributed Systems Management." *IEEE Transactions on Software Engineering,* 25**,** 852-869.

Microsoft (2007) "Microsoft Surface." *www.microsoft.com/surface*

Nath, B., Reynolds, F. & Want, R. (2006) "RFID Technology and Applications." *IEEE Pervasive Computing,* 5**,** 22-24.

Nicolai, T., Resatsch, F. & Michelis, D. (2005) "The Web of Augmented Physical Objects." *Proceedings of the International Conference on Mobile Business (ICMB'05).*

Want, R. (2006) "An Introduction to RFID Technology." *IEEE Pervasive Computing,* 5**,** 25-33.