Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

L-SIM : SIMULATING BPMN DIAGRAMS WITH A PURPOSE BUILT ENGINE

Anthony Waller
Martin Clark
Les Enstone

Lanner Group Ltd.
Clews Road, Redditch
Worcs. B98 7ST , UK

ABSTRACT

This paper describes the experiences and technical challenges encountered by the Lanner group in building a Java based simulation engine to simulate BPMN diagrams. It highlights the difference between conventional discrete simulation tools and the requirements for the BPMN engine. In particular the need for implicit rather then explicit queuing, the complexities of the patterns of parallel token flow and problems of valid but infeasible diagrams are discussed. Further the limitations of the BPMN standard for producing useful simulation models are highlighted, in particular the availability of resources.

1 INTRODUCTION

Business Process Modeling (BPM) tools are migrating towards a new emerging standard notation called Business Process Modeling Notation (BPMN). Full details of this standard are available on the BPMN web site. This paper describes the experiences and technical challenges for the Lanner group in building a simulation engine, L-SIM, for simulating these BPMN diagrams. The Lanner group was formed 10 years ago by a management buyout from AT&T. Lanner has a long and successful history in both the use and development of simulation technology. Its SEE WHY tool was the first commercially available visual interactive simulation package and WITNESS was the first of the industrial strength 4GL simulators.

BPMN is an emerging standard for producing diagrams for Business Process Modeling. The applications include Business Process Re-engineering and system integration. Its forerunners include the IDEF methodologies.

The aim of the L-SIM tool is to bring the benefits of simulation analysis to BPM projects. It enables:

- The production of a simulation model from a BPMN diagram (and variants of BPMN)
- The production of an animated visualization of the diagram as it would be operated or occur
- The production of quantitative statistics such as the utilization of the components of the diagram

What it does not provide is a BPMN drawing tool, as there are already many very good diagramming tools available in the marketplace.

Lanner has developed technology/OEM partnerships with Telelogic and IDS Scheer and Lanners Simulation Solutions are embedded in both these company's BPM Products.

2 WHAT IS BPMN?

2.1 BPMN Overview

The Business Process Management Initiative (BPMI) has developed a standard Business Process Modeling Notation (BPMN). The primary goal of BPMN is to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes. Thus, BPMN creates a standardized bridge for the gap between the business process design and process implementation.

Another goal, but no less important, is to ensure that XML languages designed for the execution of business processes can be visualized with a business-oriented notation.

An example of a typical BPMN diagram, taken from the BPMN specification, for the process of a doctor attending to a patient is shown in Figure 1. This illustrates some or the core concepts of BPMN.

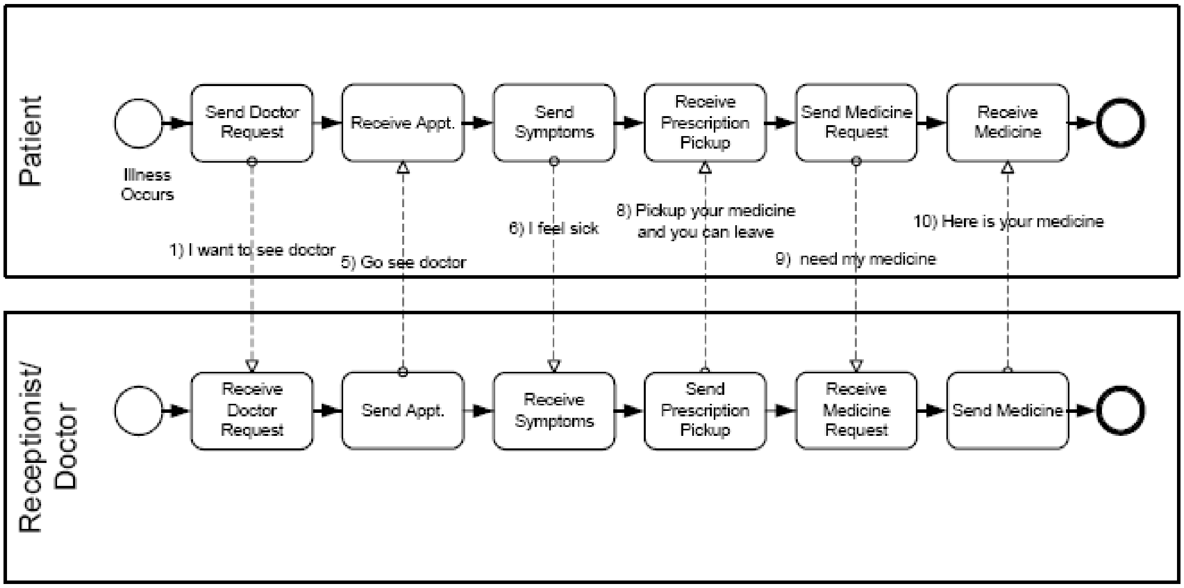


Figure 1: Example of a BPMN Diagram

In BPMN parlance an illness for the patient processes and a case for the doctor and staff processes are represented as tokens. The circular symbols represent events. Tokens are considered to be created at start events and follow a sequence flow through activities (of which tasks are the most important type) until they reach an end event. At these points the illnesses and cases are considered finished. The doctor activities and patient activities are kept in separate pools and communication between doctor and patient are represented by message flow. Note that patients and doctors together with their staff are not represented explicitly as separate objects in the diagram.

3 BPMN AND SIMULATION

The BPMN diagramming standard is part of a wider Business Process Management Initiative (BPMI) for the definition, design, deployment, execution, control, and monitoring of systems to manage business processes. In concept parts of this overall BPM initiative align closely to what we, in the simulation community, have been practising and preaching for many years. Figure 2 illustrates the BPMN cycle, together with Lanner's vision of where simulation and its related technologies bring value and benefit to BPM.

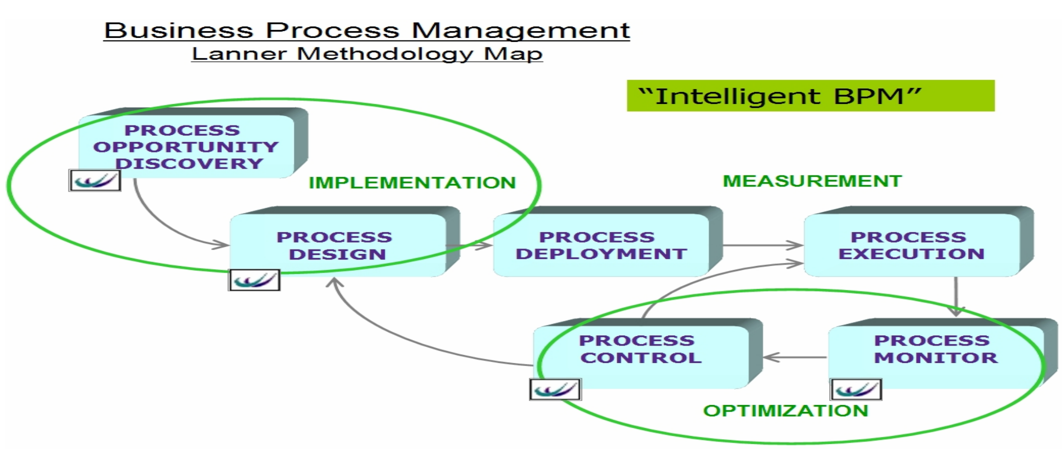


Figure 2: Intelligent BPM

Lanner believes there are two different potential users of simulation within a BPM environment that align closely with the BPM implementation process steps:

- The process Designer for Process Discovery and Design.
- The process Manager for Process Monitor, Control and improvement/optimization

Lanner believes that there is a potential symbiotic relationship between BPM and simulation. Experience in modeling techniques and qualitative and quantitative understanding gained by building and running simulation models add considerable value to BPM projects. BPM projects bring the benefits of data capture and organisational acceptance, often bugbears of traditional simulation modeling projects.

Recently, Gartner, the leading research and analysis provider, stressed that simulation and optimization can be seen as a “business core competence” to support BPM and Business Process Improvement (BPI) (Gartner 2006). This reinforces the Lanner stance that simulation be taken seriously in BPM suites, building on the extensive knowledge bank generated by practitioners in simulation across a wide variety of industries and applications.

4 SIMULATING BPMN DIAGRAMS

4.1 Overview

This paper concentrates on the first of the intelligent BPMN application areas, which most closely relates to traditional simulation modeling. Within the process Discovery and Design environment, the BPM process design solution is linked to the L-SIM simulation engine via a BPMN diagram. Core to L-SIM is a set of objects that map directly onto the symbols of BPMN diagrams. The host application captures the information required for simulation via these objects, and communicates to L-SIM via a BPMN compliant API. Results from the execution of the resulting simulation model are communicated to the host for intuitive and efficient feedback. The results may be both immediate state feedback of the model state to drive animated process visualization, and quantitative, end of run, results of model components. Figure 3 illustrates the overall L-SIM architecture.

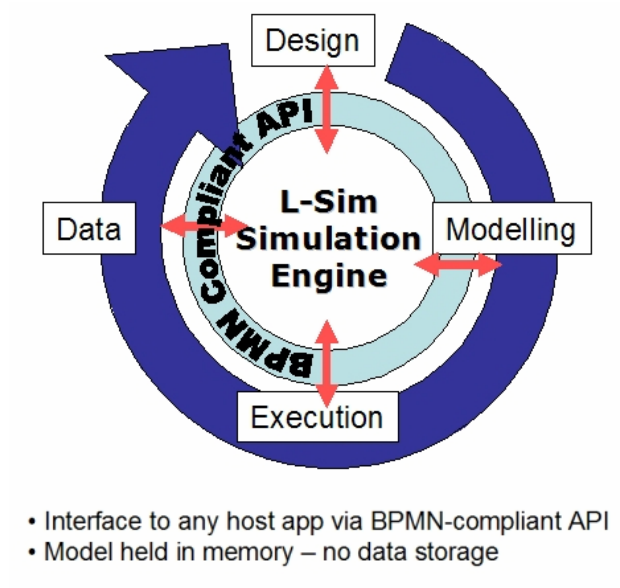


Figure 3: L-SIM Architecture

4.2 Design Concepts

At a conceptual level the majority of discrete event simulation tools work in a similar way. Entities / transactions / parts move through a series of queues / lists / sets / buffers acquiring and releasing resources / machines / labor as they move through the model domain. The entire model is driven by a sequence of discrete events which occur when an activities / processes / delays / cycles are completed, and the entity movement that occurs as a consequence of these events occurring.

A superficial view of BPMN diagrams suggests that conceptually this is similar to a simulation language such as GPSS or SIMAN. Instead of transactions or entities moving between blocks, we have tokens moving between symbols. However it quickly became apparent that this is not the case. In simulation languages the modeler must include queues where the transactions or entities wait before acquiring resources or messages, or even in a simulator the modeler must define these queues. In BPMN there is no corresponding equivalent to queues. However analysing a BPMN diagram from the perspective of generating a simulation model, it is apparent that tokens must wait, or queue, at various points in the diagram. For example in BPMN a token may be routed from one task to another. Each task may be given a time to complete. If the token at the task requires a resource (or a performer in BPMN speak) to complete the task, it is clear to us in the simulation community, that the required resource may not always be available, and that the token will have to queue for that resource if it is not available, leading to time to complete the task being longer than specified. Further, BPMN specified that tokens may be split and merged in a very prescriptive manner. This results in tokens that must wait, or queue, to be matched with their corresponding tokens.

4.3 Queuing

Traditionally, simulation solutions tend to start with queues and then define the activities that move entities between them. From a casual exploration of symbols in a BPMN diagram, you might be forgiven for believing it is simple to identify where queuing may occur. However on closer inspection, there are many complexities hidden within the interpretation of a BPMN diagram. The methodology hides many opportunities for queuing, and considerable challenges for the simulation engine to efficiently progress entities through the model.

Furthermore, when adding simulation to a BPMN modeling tool, we must be acutely aware that the user is a business modeler and not a seasoned simulation modeler. L-Sim development has taken the approach that: a BPMN simulation tool must match the formal specification for diagrams, and simplify all simulation issues for the user. The subtleties of modeling queues must therefore remain a challenge for the simulation development team and not the BPM modeler.

Some of these complexities of queuing patterns that arise from the routing patterns of splitting and joining token flow through gateways and synchronization of message flow between tokens in BPMN diagrams, are discussed below.

4.4 Gateways

The main ways of controlling the flows of tokens in a BPMN diagram are gateways. A gateway may control either the divergence of, or convergence of multiple sequence flows. The types of gateway for both divergence (split) and convergence (merge) of flows include:

- XOR – exclusive decision and merging. A single token arriving at an XOR split gateway will take one, and only one, of the possible output flows. Similarly only one token arriving at an XOR merge gateway will be sufficient to trigger the output flow.
- AND – parallel routing. A single token arriving at an AND split gateway will trigger token flows down ALL the possible outputs. Similarly an AND merge gateway requires a token to arrive at all input flows before triggering the flow of a single token on the output.
- OR – inclusive decision and merging. A single token arriving at OR split gateway will generate token flows down any of the output flows of the gateway providing the condition on that output flow is met. Note that a peculiarity of the specification dictates that at least one token must be output through, typically, a default route. Thus if an OR gateway has 6 possible outputs, then a parallel flow of between 1 and 6 tokens will result. An OR merge gateway works in the corresponding way and will collate the required number tokens arriving at the gateway before routing a token on the output flow. One of Lanner's partners likes to prescribe an OR junction with a probability of a token going down each leg. For example they may specify 80% of tokens to go down one leg and 40% down another. However if we naively interpret this data, clearly there is a probability of (1-0.8) * (1-0.4) or 0.12 of no tokens being generated. It is currently an open question as to whether there should be an implicit default third leg down which a token is sent and destroyed, or whether additional data is needed on the exact breakdown of the probabilities of each potential combination, or even whether we should resample in the zero token case until at least one token is dispatched from the gateway, resulting in more tokens than originally anticipated.

Usually gateways are used in matched pairs of split and merge gateways, where the control logic on the split gateway is the same as that on its corresponding merge gateway, as shown in Figure 4. The first of the diagrams, with matched XOR gateways is known as a workflow simple merge. A token arriving at the XOR split gateway will be routed down one of the two legs, and when it arrives at the merge gateway it will be routed downstream. This is relatively straight forward to implement.

However the second two diagrams, with matched AND or OR gateways (known as workflow synchronizing joins) are more complex. One or two tokens may be routed from the split gateway. The merge gateway is then used to recombine the same number of tokens as emanated from the split gateway, before being routed downstream. This will always be two for the AND case and may be one or two for the OR case. Further the specification dictates that tokens are matched at the merge with only the same, or corresponding, tokens that were created at the split at the same time as itself. Figure 5 illustrates the matching pattern that must be obeyed. Initially a token A arrives at the split and creates two tokens A1 and A2. Subsequently a second token B arrives at split and creates tokens B1 and B2. Token A1 completes its task and arrives at merge gateway and waits, and then token B2 completes its task and arrives at the merge. Token A1 and B2 are not combined. Token A1 must wait for token A2 to arrive, and these will continue as the single token A. Similarly token B2 waits for its correspondent B2 to arrive, before they continue as the single B2.

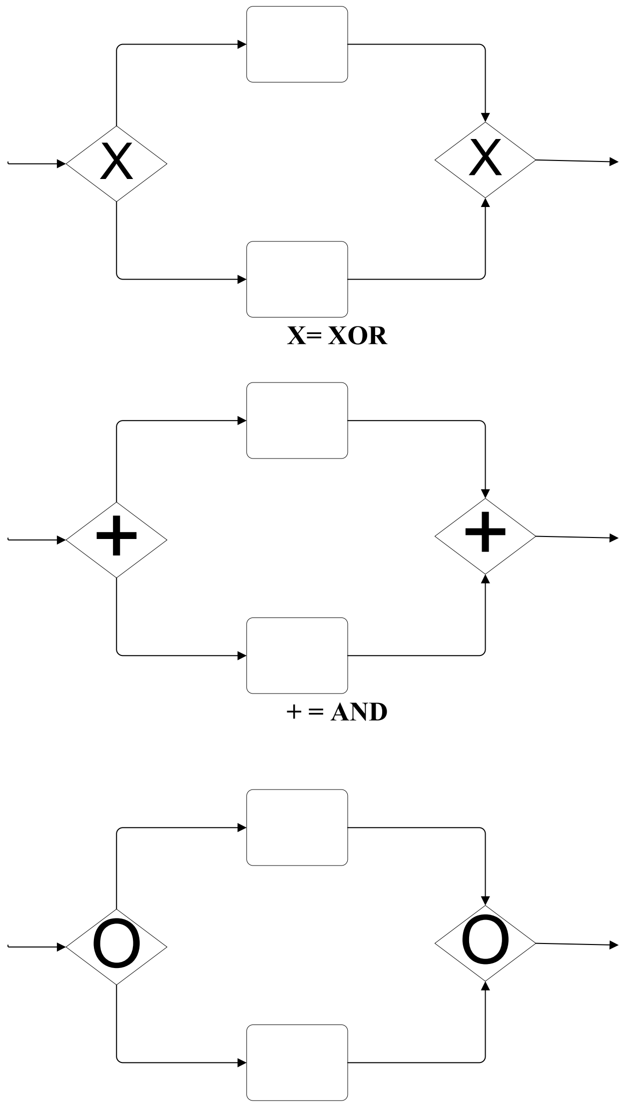


Figure 4: Matched Gateways

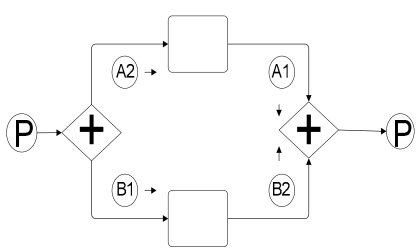
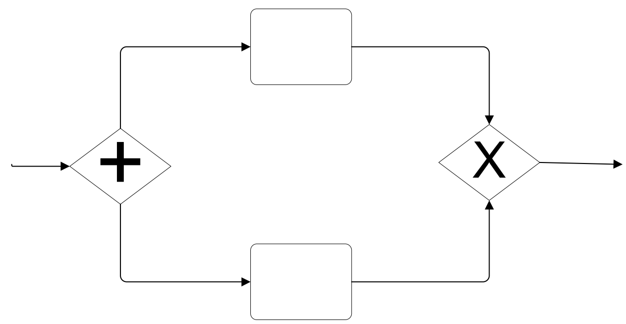


Figure 5: Matching Requirements

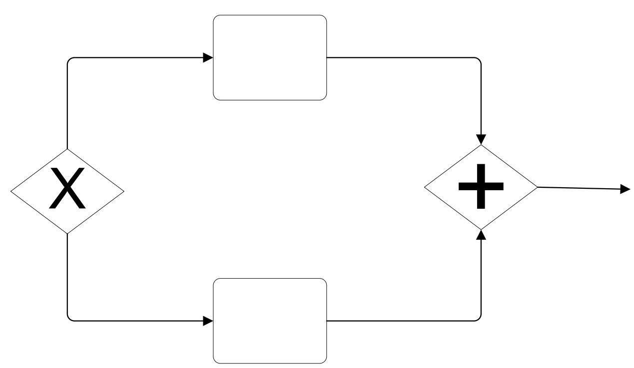
As well as using gateways in matched pairs, it is possible in BPMN to mix and match gateway types. For example an AND merge gateway may be matched as illustrated in Figure 6. This combination is known by the Workflow Patterns Initiative, a successful and much referenced web based repository of such structures, as a Workflow Pattern Discriminator (Workflow Patterns Initiative web site). The behavior specified for such a combination is that two corresponding tokens will be generated by the AND gateway. The first of these that arrives at the XOR junction will is routed downstream as soon it arrives. However the second token will be destroyed when it arrives at the XOR junction.



+ = And , X = Xor

Figure 6: Workflow Pattern Discriminator

However not all possible combinations of split and merge junctions make sense. For example, if we reverse the above diagram and match an XOR split gateway with an AND merge gateway as illustrated in Figure 7. A deadlock will result. Tokens are routed down either the top or bottom leg, and will then stay at the AND junction because their corresponding tokens have not been generated. The AND gateway will not be able to route on the singleton tokens that do arrive. An intelligent special purpose BPMN drawing tool would be able to identify such problems. However BPMN diagrams may be drawn in any drawing tool, and as such the L-SIM tool will identify the deadlocks at runtime.



+ = And , X = Xor

Figure 7: Infeasible Pattern

4.5 Messages

In the Doctor example (Figure 1) messages are sent from patient to doctor and from doctor to patient. There are times when the patient illness tokens have to wait to receive messages from the doctor case tokens, and (conceivably) the doctor has to wait for messages from the patient. The handling of messages thus requires implicit queuing of both tokens waiting for messages and messages waiting for tokens to get to the stage where they can accept the message. Further it is important for tokens to accept the message that is intended for it, rather than another token.

4.6 Implementation

The exact details of L-SIM's implementation remain confidential. However the token and message objects are far more intelligent than the equivalent Part objects in Lanner's WITNESS package (and presumably other conventional discrete event simulation tools too). As well as supporting attributes, as in any good simulation package, tokens are aware of the status of the correspondents to enable them to be efficiently matched or deleted at merge gateways, in the appropriate manner.

The L-SIM engine was initially written in Java to facilitate integration with Lanner's partner's tools. Initial internal misgivings about using Java for a simulation tool after a long history of using compiled (Microsoft) languages ranging from Assembly language, through Fortran, C and currently C++, were misplaced. Java with its Object Orientated capabilities provided an excellent tool for implementing the design patterns we have had to adopt to mimic the subtleties of simulating BPMN diagrams. Run speeds are proving to be comparable to Lanner's WITNESS product for simple models and far superior when the full complexities of the BPMN approach are used.

L-Sim is also available in C# for integration with .Net based applications.

5 EXTENSIONS TO BPMN

Lanner firmly believes that simulation technology can add considerable value to business process modeling activities. The ability to test processes and the ability to visualise them, before they are implemented add considerably to their understanding and will provide scope to identify opportunities for their improvement, especially as the behaviour demanded by the diagrams can be subtle or erroneous. Further the collection of quantitative statistics on the performance of the system being modeled can be used to properly plan its implementation.

However the main drivers in the evolution of BPMN are (1) to provide a universal diagramming notation for business processes and (2) to build IT systems on the processes so defined. These drivers are generally not the same as those required in a simulation modeling project. The BPMN specification clearly states, that as yet, it does not include the modeling of resources. Lanner's vast experience in simulation modeling is that availability patterns of resources due to stochastic effects such as equipment breakdown, and planned effects such as worker availability through shift patterns are key features in the majority of simulation models in its conventional application domains such as capacity analysis and planning, and schedule analysis.

L-Sim provides the capability to assign multiple "resources" to any task, rather than just "performers" to "user tasks". By providing a full resource model, shift model and calendar based recurring events system; L-Sim extends the BPMN specification considerably.

As BPMN is adopted more by BPM vendors, there is a window of opportunity to influence the data capture requirements of BPMN modeling tools to allow such tools to benefit from the validation, process improvement and optimization simulation can provide.

BPMN would certainly be enhanced if the following L-Sim Features were adopted into the next release of the BPMN specification:

- The ability to use resources other than human resources or performers on tasks. Human resources will generally not be the only resource required to perform a task. Other scarce resources such as specialised equipment may also be needed.
- Prioritisation of resource requests. Experience in simulation modeling tells us that when a resource has a choice of more than one outstanding task to perform, it must make a choice of which task to do first. This may be on a first come, first served basis but often one task will be more important than the other and a mechanism is needed to prioritize requests for resources. Note that the current BPMN specification does not include any references to priorities.
- Pre-emption of resources. Further to the above some tasks may be so urgent that they require resources to be taken from, or pre-empted, from tasks which they are currently servicing.
- The ability to specify more than one resource for a task. The BPMN specification does specify that multiple performers may be allocated but is not prescriptive in how this should be achieved, and this could be a combination of human resources and special equipment.
- The ability to specify alternative resources. Again experience tells us that there will generally be more than one type of a resource available, and

there will often be a choice of which resource type to use to perform a particular task.

- The ability to vary the level of resources over time in a planned manner. For human resources, planned staffing levels will often vary during a working day or vary from day to day. Equipment required to perform tasks may become unavailable through, for example, planned maintenance.
- The ability to vary the level of resource availability in an unplanned or stochastic manner. The capacity of a system can be considerably affected through unplanned effects, that we typically model using random or stochastic effects. Examples include staff illness, equipment failure or a computer crash.
- Inter-task times. After completing a task or set of tasks a resource may not be able to undertake its next task immediately. For example a human resource may be required to take time to walk between tasks and equipment may require servicing after completion of a number of tasks.

These are all effects that can have important effects on the capacity and operational characteristic and performance of the real life system being modeled. We, in the simulation community, can bring our considerable experience in practical modeling considerations to provide the BPM community with a more complete and accurate model of a business process. The modeling of resources has long been a strong point of traditional simulation modeling tools, and Lanner believes that it can provide value to the BPM community with the ability to include these extensions to BPMN.

6 SUMMARY AND CONCLUSIONS

The L-SIM simulation tool provides a tool that can link the benefits of BPM and simulation modeling. The qualitative and quantitative understanding of the performance can add considerable value to the Process Discovery and Process Design phases of a BPMN project. The BPMN diagramming standard provides a bridge between the two domains. However, because of its formal nature, sequence flow and message flow that can arise from BPMN diagrams can potentially be much more complex and subtle than expected. This provides a challenge to the simulation modeler (or engine designer) to match the specified behavior patterns. However, these complexities and subtleties emphasize the need, or at least benefit, of simulation modeling to properly understand these effects. Further the omission of the specification for the modeling of the details of resource availability provides an opportunity for the simulation community to progress the evolution of the BPMN standard to handle the modeling of the practical implications of resource based constraints.

REFERENCES

BPMN web site. Available at <http://www.bpmn.org>.

Gartner. 2006. *Top Questions about Business Process Improvement (BPI)*, Gartner Group, Inc.

Workflow Patterns Initiative web site. Available at <http://is.tm.tue.nl/research/patterns>.

AUTHOR BIOGRAPHIES

ANTHONY WALLER received a BSc (Hons) in Statistics from the University of St. Andrews in 1981 and has worked in the field of Operational Research and particularly simulation ever since. He has worked on many diverse international assignments as a consultant and is currently the Director of Products at Lanner Group.

MARTIN CLARK studied for a Ph.D. in theoretical elementary particle physics at Manchester University. He has been employed by the Lanner Group and its predecessors since 1977 where he was the creator of the original WITNESS simulation package. He is currently working on the advanced development of a number of innovative simulation capabilities.

LES ENSTONE has worked for a variety of software development companies and has specialized in the development of simulation and advanced scheduling solutions. Currently he is the Chief Technology Officer for Lanner Group.