

# Picture Archiving and Communication System Design Issues: The Importance of Modelling and Simulation

Wim J.J. Stut, Jr, Maarten R. van Steen, Luul P.J. Groenewegen, and Albert R. Bakker

Picture archiving communication system (PACS) development turns out to be very complex. Due to both the vast amount of data and the complexity of hospital organizations, currently only small-scale systems have been realized. And although the experiences obtained with these systems are essential, there is a risk for underestimating the complexity and requirements inherent in hospital-wide PACS systems. In this paper, it is advocated that modelling and simulation be used as tools to obtain insight into the behavior and structure of future PACS systems. Modelling and simulation can also be used to actively support the design of PACS, especially its software. In order to capture the full complexity of PACS in a simulation model, and to take full advantage of simulation as a design tool, the development of a new modelling method has begun. This method is based on semantic data models and decision processes, and can be used for both system analysis and design. The first systems modelled with this method were imaging procedures in a hospital and a computer network. The resulting simulation models are a direct reflection of reality, and have a high degree of modularity. Consequently, in spite of the complexity of the systems, their models are easy to understand and maintain.

© 1990 by W.B. Saunders Company.

**KEY WORDS:** PACS design, performance prediction, simulation, model construction.

**A** DIGITAL SYSTEM for the storage, transportation, presentation, and processing of medical images within a hospital (a so-called picture archiving and communication system, or PACS) is expected to be realized within 1 or 2 decades. The claimed advantages of the PACS concept include a better accessibility of images, reduction of image handling and recording

costs, and improved diagnostic quality by applying sophisticated image processing techniques. BAZIS (the central development and support group hospital information system) has been actively involved in the PACS field for quite some time now. To support the development of hospital-wide integrated image information systems, in 1984 BAZIS initiated its image information system (IMAGIS) project. In close cooperation with physicians, industry, and other PACS research centers, IMAGIS focuses on PACS research and software development.

## *Motivation for PACS Simulation*

Worldwide, the realization of PACS systems is still in its initial stages. Although the results of some first small-scale experiments are promising,<sup>1</sup> the realization of large-scale systems appears to be difficult. The vast quantity of image data leads to storage problems and unacceptable waiting times. Furthermore, the changeover to clinical working methods based on digital images causes significant organizational problems. To assist in solving these problems, BAZIS decided to use modelling and simulation techniques as a means to support the laborious PACS design process.<sup>2</sup> These techniques are beneficial in three ways.

First, applying simulation techniques can substantially contribute to obtaining insight into a system's behavior in an early stage of its development. Without simulation, it is difficult to foresee what the full effect of an early design decision will be with respect to the final system. This is certainly the case when considering performance issues, and it turns out that even in the general case of information system development, a complete re-engineering is often needed if initial behavior requirements are to be met.<sup>3</sup> By building small-scale PACS prototypes, substantial feedback from practice has been obtained. Although this has contributed to the development of systems that have been evaluated against actual functional requirements, the performance of full-scale systems (which have yet to be built) remains to be seen. Simulation, on the other hand, is an excellent tool for extrapolating the practical experiences obtained at small-scale

---

*From BAZIS, Central Development and Support Group Hospital Information System; TNO Institute of Applied Computer Science; and the Department of Computer Science, Leiden University, The Netherlands.*

*Supported in part by the Netherlands Organization for Scientific Research (NWD).*

*An earlier version of this paper appeared in the proceedings of Computer Assisted Radiology '89.*

*Address reprint requests to WJJ Stut, Jr, BAZIS, Central Development and Support Group Hospital Information System, PO Box 901, 2300 AX Leiden, The Netherlands.*

© 1990 by W.B. Saunders Company.

8097-1889/90/0304-0001\$03.00/0

experiments to a hospital-wide PACS, thereby reducing the possibility that we underestimate the associated complexity and the consequences of the vast amount of data.

Second, simulation can be used to actively support the design of software systems, as both the software and its environment (such as users and hardware) can be reflected in a model. The model part that specifies the software can now be considered as a prototype version of the eventual system. By gradually replacing its components by fully developed software modules, this model part is incrementally transformed into the eventual realization. As a last step, this realization can be transferred from its simulated to its concrete environment.

The applicability of this form of software development follows from the observation that modelling software for future PACS systems is in fact the equivalent of designing this software. For example, when modelling an image-prefetching algorithm,<sup>4</sup> the modeller has to specify explicitly the events in the hospital (or its information system) to which the algorithm should react, and has to design routines to handle these events. After the behavior and the effects of the prefetching algorithm have been evaluated by means of simulation, it can be used as a starting point in the derivation of the actual software.

Finally, the use of simulation can be extended from the design phase into the day-to-day operation of a PACS. After all, the model that has been used to predict the effect of design decisions can also be used to answer so-called "what if" questions in the operational phase. It can act as a testbed that is easy to control and modify without affecting the users, and eliminating the need for time-consuming, expensive, and difficult real-world experiments.

#### *Requirements on Model Construction*

Building simulation models of large and complex systems such as PACS requires that attention be paid to the following aspects. First, the construction of the simulation model should be supported by formal modelling techniques, yielding consistency, completeness, and avoidance of ambiguity. Second, to manage the complexity of the simulation model, it is necessary that the model be easy to modify, reliable, and understandable. These requirements are more easily met

when the designer of the model uses software engineering principles like data abstraction, information hiding, and modularity.<sup>5</sup>

If simulation is also to support the actual design of systems, special attention must be paid to the construction of the model itself, for this directly influences the development of the actual system. It can be expected that this goal is more easily achieved if the model not only captures the behavioral aspects of the system, but the structural ones as well. In this case, the organization of the model will strongly resemble that of the system. In other words, the internal mechanisms and architecture of the system are reflected in a more or less direct manner by the simulation model. By manipulating simulation objects that correspond to objects in the system, system design can then be simultaneously addressed.

Unfortunately, in the field of simulation modelling, most formal techniques result in an abstract model that concentrates solely on the behavioral aspects of a system. Examples of these models often occur in the field of continuous simulation, or in those situations in which systems are modelled as queueing networks or Petri nets.<sup>6</sup> Although these models can be used to study the dynamics of a system, they do not capture structural aspects.

Available software engineering methods bring no relief either. Many methods support only one phase in the traditional software development life cycle. For example, many system analysis methods focus on the specification of a system's external behavior, but do not specify its internal structure (eg, what processes there should be, how their communication is synchronized, etc). On the other hand, structured design methods often emphasize the internal structure of a software system but explicitly presume a requirements specification. When developing PACS, both analysis and design are essential.

On the other hand, integrated software engineering methods that do attempt to cover more phases of the life cycle have often been tailored towards a specific class of systems. For example, one can distinguish between methods for development of information systems that are centered around a data base, and methods that concentrate on real-time system development (in which system behavior plays a prominent role). Unfortunately, systems that have many different kinds

of characteristics, such as PACS, often lack a suitable development method. Furthermore, existing software engineering methods head straight towards the final system realization, and are thus seen not to support the approach of design by simulation.

To summarize, in order to allow simulation-based PACS development the modelling method needs to meet two requirements. First, it must be an integrated modelling method that supports both system analysis and design, addressing both static and dynamic aspects. Second, the method must simultaneously support the development of a system realization, and the development of a simulation model. Because existing system development methods do not meet these requirements, at BAZIS the development of a new modelling method was initiated.<sup>7,8</sup>

### MODELLING METHOD

#### *Overview and Background*

Our modelling method comprises three phases. In each phase, a different means for system specification is used: semantic data models, parallel decision processes, and a programming language. A schematic representation of the modelling method is found in Fig 1.

Model construction is started with identifying and structuring the data, facts and real-world components (called objects) that play a role when considering a PACS in a hospital. Examples of objects are radiologists, technicians, examination requests, and images. For this purpose, a semantic data modelling method has been devised.

In the area of conceptual data base design, data models are used to express the data's logical structure and use.<sup>9</sup> Since the mid-1970s, semantic data models (of which the Entity-Relation-

ship model is considered to be one of the first) have been proposed, allowing data base designers to think of data in ways that correlate more directly to how data actually manifest.<sup>10</sup> With regard to traditional data models, semantic data models provide higher level concepts by defining a more abstract level of system description. An example of such a concept is generalization, by which similar objects are abstracted into a single, higher-level object. The resulting so-called ISA relationships are a powerful means to express a system's hierarchical structure. Moreover, by using inheritance, properties of the general object can be passed on to the more specific objects.

Semantic data models on their own are not appropriate (nor intended) to make a simulation model of a PACS. They focus on the structure of objects, ie, their static properties and relations, but do not easily allow for the notion of time to be incorporated. Therefore, special facilities have to be provided to model the dynamics of objects and the communication between objects (ie, their behaviors).

In the literature, several formal techniques for modelling a system's dynamics can be found. The disadvantage of most of these methods is that the resulting models do not directly reflect the dynamics of the real system (ie, they are too far from the human conceptualization of the problem domain). Net-based formalisms such as Petri nets do improve this situation, but by nature often lack sufficient support of a modular setup of models. A high-level formalism that does support modularity and hierarchical organization of models is Paradigm.<sup>11</sup>

The Paradigm modelling method is based on parallel decision processes. The advantage of this method is that the resulting model is a highly modular map of the system studied, of its sequential components, and of their communication. Unlike Petri nets, the components are easy to distinguish. Moreover, it is easy to adapt the model, to add or refine modules, or to add processes in order to control more complex communication.

Hence, succeeding the semantic data modelling method, we use Paradigm to develop a more detailed system specification. In fact, the semantic data model is partially transformed into a Paradigm model, and together with this transformation more details can be added. An important

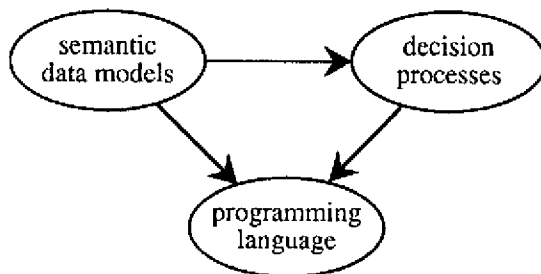


Fig 1. The three phases in the modelling method. The arrows reflect the transformations of system specifications.

part of these details are time aspects, since they further specify the dynamics of the system.

After a PACS has been specified in terms of semantic data models and Paradigm, modelling continues in a third phase, consisting mainly of adding more details, and now using a programming language as a means for model specification. This eventually leads to a realization of the required simulation model. For this purpose, we are currently using the object-oriented programming language Simula, whose class construct can be used to implement both the static properties and the behavior of an object in an elegant way.

In the following sections, the semantic data modelling method and Paradigm are explained informally. For a more complete description of the method, including the transformation of system specifications, refer to Stut et al.<sup>8</sup>

### Semantic Data Models

The semantic data modelling approach comprises three main structural components: objects, functional relationships and ISA relationships. Decomposition of a model into smaller parts, is supported by means of so-called fragments.

**Objects.** The basis of a semantic data model is formed by objects, used for modelling the components of a system. Objects are divided into passive and active objects. Passive objects differ from active objects in that the latter display autonomous behavior. It is emphasized that the choice to model a component as either a passive or an active object strongly depends on the modeller's view on the system. Typical examples of components that could be modelled as active objects are patients and technicians, whereas examinations and images may possibly be modelled as passive objects. In the visual representation of a model, (see Fig 2) objects are drawn as

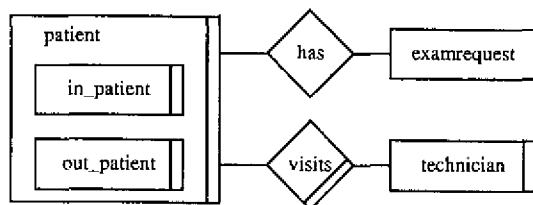


Fig 2. A simplified example of a semantic data model. The active object *patient* is involved in a passivity relationship *has* and an activity relationship *visits* with the objects *examrequest* and *technician* respectively. Both relationships are inherited by the *patient's* subobjects *in-patient* and *out-patient*.

rectangles with active objects having a double-lined side.

**Functional relationships.** Objects are connected or associated by means of functional relationships. These relationships are divided into activity relationships conceptually describing dynamic properties of active objects, and passivity relationships reflecting the static properties of passive and active objects. Functional relationships are visualized as diamond-shaped boxes with activity relationships having a double-lined side.

**ISA relationships.** ISA relationships provide the basis for the use of inheritance. When using inheritance in the definition of an object, a designer need only specify what is new about the object in comparison with the objects from which it inherits properties. ISA relationships are visualized as follows: the rectangle representing the object that inherits properties is enclosed by the rectangle representing the object that provides these properties.

**Fragments.** Instead of constructing a large model containing many objects and relationships, we model a system by means of several fragments. A fragment is a semantic data model representing the view of one element on the real world (or actual system). This element can be either an object or a relationship and is called the central element of the fragment. All other objects and relationships in a fragment are denoted as virtual objects and virtual relationships, and correspond to actual objects and actual relationships respectively.

To illustrate, consider an imaginary radiology department at which each examination procedure is performed by a technician, after which it is authorized by, possibly, another technician. The view of the object *examination* on the real world could be represented by the fragment of Fig 3.

It is important to note that different virtual objects may represent the same actual object. So, when in the above example an examination is done and authorized by the same technician, both *exec\_technician* and *auth\_technician* represent this technician.

Virtual objects in the fragment of a relationship represent the (unique) role of a real world object with respect to that relationship. For each activity relationship we distinguish one special

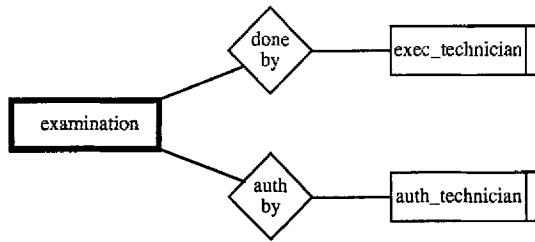


Fig 3. A fragment with central object *examination*. Its virtual objects *exec-technician* and *auth-technician*, respectively represent the technician who performed the examination procedure and the technician who authorized it. Primary element in bold lines.

role, namely the role of the object that actually effectuates the relationship. This object, known as the agent object of an activity relationship, is denoted by means of a directed edge as shown in the fragment of the relationship *selects* of Fig 4.

We can now describe fragments of objects in an unambiguous way by labelling the edge between an object and a relationship with the role that the object has with respect to the relationship (see the relationship *selects* in Fig 5 for an example).

An example that will be used in the next section describes a radiologist at an imaginary radiology department. After an examination procedure has been performed and authorized by technicians, it is added to a report list of a radiologist. By using this report list, each radiologist knows which examinations have to be studied and reported. In 20% of the cases, the radiologist wants to have some background information about the examination procedure: in that case, the technician who performed this procedure is called. The fragment of Fig 5 represents the view of the radiologist on the system. Note that with respect to the relationship *selects*, *radiologist* is

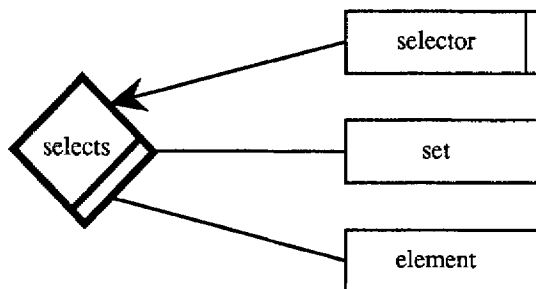


Fig 4. The fragment of activity relationship *selects* with agent object *selector*. Central element in bold lines.

the *selector*, *worklist* has the role of *set*, and *examination* has the role of "element."

The semantic data models provide global information on dynamic aspects through the use of activity relationships. However, although these relationships conceptually describe activities that can be done by active objects, semantic data models do not adequately describe the actual dynamics of a system. Examples of these dynamics are the order in which activities take place, and behavior dependencies between objects. For that purpose we use Paradigm.

### Parallel Decision Processes

In Paradigm, a system is modelled as a collection of decision processes which together constitute a so-called parallel decision process. Each decision process is used to model a system component that essentially exhibits sequential behavior. When modelling a radiology department, examples of decision processes are *radiologist*, *technician*, *secretary* and *patient*.

A decision process has a set of states, each state reflecting the relevant properties of a system component at a particular moment. The changes that occur in a component are reflected in the associated decision process by state transitions. A decision process is represented by a digraph, in which a node represents a state, and a directed edge represents a state transition. As an example, in Fig 6 we present a decision process for the radiologist mentioned above.

The interpretation of the sojourns in the states is (1) selecting an examination from the worklist; (2) studying the examination; (3) calling the technician who performed the examination procedure and waiting until he or she has arrived; (4) consulting the technician, and (5) making a report.

The transitions between the states are  $1 \gg 1$  the worklist is empty;  $1 \gg 2$  an examination has been selected from the worklist;  $2 \gg 3$  the radiologist wants to consult the technician;  $2 \gg 5$  the radiologist does not want to consult the technician;  $3 \gg 3$  the technician has not yet arrived;  $3 \gg 4$  the technician has arrived;  $4 \gg 5$  the consultation has ended; and  $5 \gg 1$  the report has been made.

In each state of the decision process, a transition is selected. This selection may be determinis-

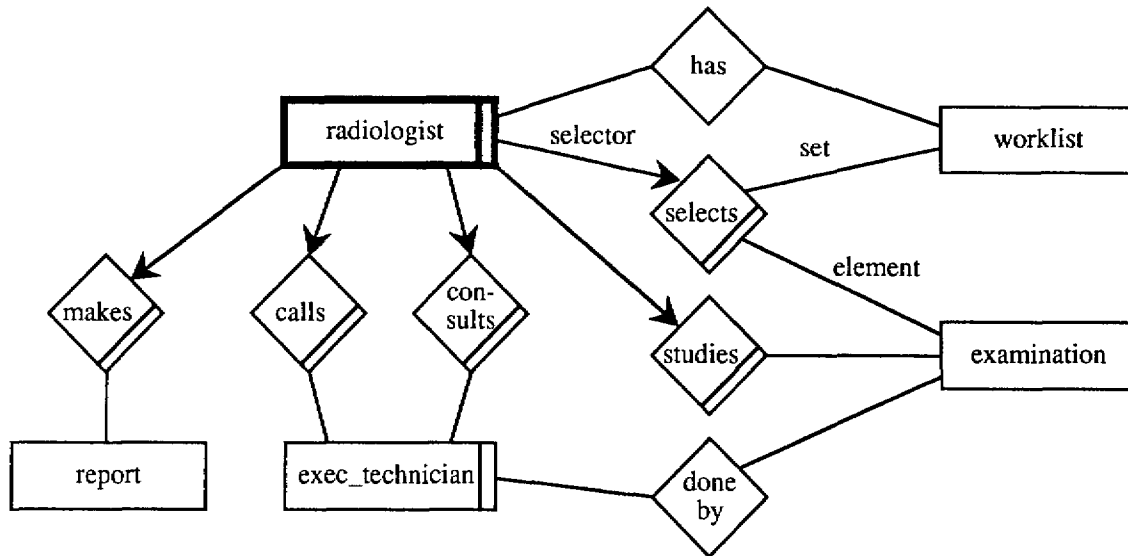


Fig 5. The fragment of the object *radiologist*. Primary element in bold lines.

tic, stochastic, or may depend on other system components. A deterministic selection corresponds to an unambiguous prescription of the next activity. In order to be able to model a system whose exact behavior is not known in advance, Paradigm offers a stochastic transition selection mechanism. Finally, a transition selection that depends on other system components is used where communication between the various model components has to be incorporated. As an example of a deterministic transition selection, in state 4 it is only possible to continue with making a report, ie, selecting the transition to state 5. On the other hand, in state 2 the transition mechanism will stochastically select the transition to state 3 with probability 0.20, and the transition to state 5 with probability 0.80. This reflects the behavior of the radiologist, who, in only 20% of the cases will consult a technician. Finally, in

states 1 and 3, the selection of a transition will depend on the availability of a next examination, and the arrival of the technician respectively.

After a transition has been selected, a sojourn mechanism determines how long it takes before this transition takes place. This so-called sojourn time roughly reflects the time needed to perform the associated activity; after it has elapsed, the factual transition takes place. For example, in state 2 for both transitions, the sojourn times reflect the time needed by the radiologist to study the examination.

When modelling a system's dynamics, the idea is not to start from scratch, but to use its static structure as expressed by means of the semantic data model as a starting point. Then, the decision processes are constructed by transforming previously established system specifications. The basis for this transformation is formed by the fragments of active objects, in particular, their activity relationships. After the derivation of the various decision processes, their cooperation is described. To this aim, Paradigm offers formal notions and model constructs, also in terms of decision processes. The way model construction in Paradigm takes place not only leads to a high degree of modularity, but also results in a uniform description of system components and the communication between them. At this place, we will not go into further details of Paradigm, but refer to Van Steen<sup>12</sup> who demonstrates that this

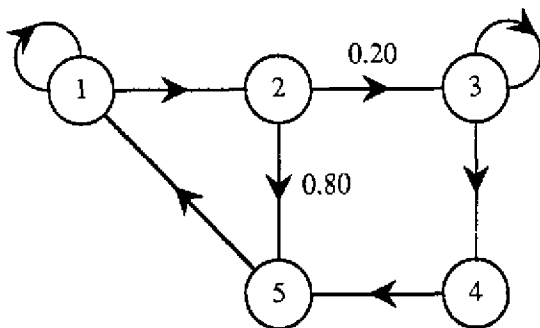


Fig 6. The decision process *radiologist*.

approach is extremely useful when modelling complex parallel systems.

### EXPERIENCES

At BAZIS, simulation has been used extensively to guide the PACS development process. An example is the simulation of a prototype PACS in the Utrecht University Hospital within the scope of the Dutch PACS project.<sup>13</sup> In this experiment, the idea was to do all imaging procedures for patients admitted at one 15-bed ward of the department of internal medicine with PACS. To this aim, all images of these patients (also those images available in the archive) were to be entered into the PACS by means of a laser digitizer. At the department of radiology, a 6-screen workstation was installed. Furthermore, images could be studied by referring physicians at a workstation at the ward of internal medicine. The storage system of this PACS consisted of a buffer and an optical disk archive.

Before the prototype PACS became operational, simulation was used to predict its performance and to evaluate different strategies for performance improvement as illustrated by the following examples.

The simulation showed that, even for this small department, storage problems would arise. By digitizing all historical images of admitted patients, the highest memory level would constantly fill up. As an answer to this, in the simulation model, a special job was started several times a day, taking care of sending images to the archive and removing them from the highest memory level. Due to the increased image traffic, however, waiting times also increased.

The fact that all users were initially connected to the same storage unit, led to interfering image requests. The simulation showed that this interference did increase waiting times as was expected. For example, the waiting times were influenced by the digitizing of historical images. With the simulation model, we could easily determine the effect of changing the moments at which the images were digitized.

The simulation showed that prefetching (or preloading) is a prerequisite for relatively short waiting times. If we are able to identify those images that have an increased probability to be requested, these images can be sent from the

archive to a higher memory level before they are actually needed, thereby reducing the delay caused by slow components such as optical disk drives. As a result, waiting times decrease significantly.

Although the results obtained with the current simulation software have shown to be very useful, some drawbacks were also revealed. One of these was that, due to the absence of an appropriated modelling method, it became increasingly difficult to maintain our simulation models. Furthermore, we felt that PACS design did not take full advantage of simulation as a design tool. In order to answer the need for a more fundamental approach, we started research into modelling methods, resulting in the development of the new method described in this article.

In spite of the fact that the theoretical aspects of the modelling method require further research, it is already used in practice. A first validation of the method has been done by modelling the working methods at the department of internal medicine of the Utrecht University Hospital. The purpose of this validation was to pass through the three phases of the method and to see whether it was applicable to a realistic situation. The results of this validation are very promising. First, the method yielded a well-structured simulation program, in which the imaging procedures were reflected in a direct manner. Second, its use had interesting side-effects: its formal character promoted a consistent and complete system specification. Besides, due to its high-level abstraction mechanisms, the combined use of semantic data models and decision processes turned out to be an excellent tool to structure an organization, addressing both its static and dynamic aspects. In other words, we expect that the method can also be used at other PACS research and development activities (such as organizational studies, user interface design for workstations, and development of image management systems).

The modelling method and simulation are currently used to support the design of a PACS network. As an alternative to the common networks, the Aachen University of Technology has conceived the fiber-optic ImNet (IMTEC Image Technology AB, Uppsala, Sweden) network. Within the scope of the hospital integrated PACS

(HIPACS) project of the European Committee, the modelling and simulation of this network is done at BAZIS.

The motivation to simulate ImNet is fourfold. First, there is a lack of experience concerning the network behavior under PACS production load. Although a prototype network has been installed, it is not yet part of a PACS that is clinically used. Second, simulation can be used to optimize the configuration of networks. So, given a particular hospital, how can we make best use of ImNet? Without changing the principles on which ImNet is based, modifying the network may turn out to improve its performance. Third, simulation can be used to enhance our understanding of the behavior of large-scale networks, which will help in the development of future networks. Finally, the presence of a prototype network and feedback from the network developers enable the validation of the simulation model and of the modelling method.

With regard to model construction, it turns out that the use of the modelling method results in a simulation model with a high degree of modularity. After the network had been specified as a combination of semantic data models and decision processes, the implementation in Simula was

a straightforward transformation. Furthermore, the fact that the model directly reflects reality increases its comprehensibility and facilitates the communication between the modeller and the network experts. Consequently, misunderstandings regarding the network's internal mechanisms can be solved in an early stage.

## CONCLUSIONS

The laborious PACS design process benefits from simulation. Several studies have demonstrated that simulation is an excellent tool for testing hypotheses and for extrapolating experiences obtained at today's small-scale prototypes to future hospital-wide systems. Furthermore, if special attention is paid to the construction of simulation models themselves, simulation can be used to support PACS design in an active manner. In order to capture the full complexity of PACS in a simulation model, and to take full advantage of simulation as a design tool, we are developing a novel advanced method for model construction. This method is based on semantic data models and decision processes, and provides high-level modelling abstractions that allow for the construction of simulation models that are direct reflections of reality.

## REFERENCES

1. BAZIS, AZU, Philips: Concluding Report of the Dutch PACS Project, Picture Archiving and Communication System. Phase I (1986-1989). March, 1990. (Available on request from BAZIS.)
2. Stut Jr WJJ, de Valk JPJ, ter Haar Romeny BM, et al: Modelling and simulation within the Dutch PACS project. *Proceedings of the SPIE (Medical Imaging III)* 1093:423-428, 1989
3. Boehm B: A spiral model of software development and enhancement. *Comput* 21:61-72, 1988
4. Bakker AR, Didden H, de Valk JPJ, et al: Considerations on an algorithm for activation of images in a multi-layered storage system within a PACS. *Proceedings of the SPIE (PACS IV for Medical Applications)* 626:637-640, 1986
5. Sommerville I: *Software Engineering* (ed 2). International Computer Science Series. Reading, MA, Addison-Wesley, 1985
6. Peterson JL: *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, NJ, Prentice-Hall, 1981
7. Stut Jr WJJ, Bakker AR, van Steen MR, et al: A semantic modelling method to guide simulation-based system development. *Proceedings 3rd European Simulation Congress, Society for Computer Simulation*: 101-106, 1989
8. Stut Jr WJJ, van Steen MR, Groenewegen LPJ, et al: An integrated approach for simulation-based system development based on transformations of system specifications. *Proceedings of the European Simulation Multiconference, Society for Computer Simulation* 655-660, 1990
9. Brodie ML, Mylopoulos J, Schmidt JW, eds: *On Conceptual Modelling*. New York, NY, Springer-Verlag, 1984
10. Hull R, King R: Semantic database modeling: Survey, applications, and research issues. *ACM Computing Surveys* 19:201-260, 1987
11. Van Steen MR, Groenewegen LPJ, Oosting G: Parallel control processes: Modular parallelism and communication, in Hertzberger LO, Groen FCA (eds): *Intelligent Autonomous Systems*. Amsterdam, The Netherlands, North-Holland, 1987
12. Van Steen MR: Modeling dynamic systems by parallel decision processes. Doctoral thesis, Department of Computer Science, Leiden University, March 1988
13. Stut Jr WJJ, de Valk JPJ, Didden HW, et al: First experiences with the modelling and simulation package MIRACLES applied to a PACS in a clinical environment. *Comput Methods Programs Biomed* 28:63-70, 1989