From RADs to DESs: a mapping from Process Models to Discrete Event Simulation

Ana I. Martínez-García[*] *and Brian C. Warboys*
Informatics Process Group
Room 2.110
Department of Computer Science
University of Manchester
Oxford Road, Manchester M13 9PL, U.K.
e-mail: { martinea, bwarboys }@cs.man.ac.uk

Abstract

This paper presents some results arising from the development of a prototype tool concerned with the automatic mapping of process models captured using Role Activity Diagrams (RADs), to the input to a Discrete Event simulator. The key element in our work is to establish, as a first step, the differences and common features between the concept domain in RADs and Discrete Event Simulation (DES) Modelling . Hence identifying the issues concerning the mapping from one to the other with the objective of facilitating the use of these kinds of linked tools. Thus, it is hoped avoiding extra effort in building the simulation models from scratch, and allowing the evaluation of proposed process model changes to be influenced by the results of a simulation study.

1 Introduction

Process models are constructed to study and to better understand processes in organisations. They are captured using particular techniques and from specific points of view [1]. Their usage ranges from, aiding understanding to automated execution. The five basic uses given by Curtis *et al* [2] are: *(1) Facilitate human understanding and communication*, by documenting and supporting procedures in organisations in a consistent and uniform manner. Creating a common ground of communication. *(2) Support process improvement*, by establishing the basis to define and analyse processes. *(3) Support process management*, giving the basis for comparing actual processes against those already established. *(4) Automate process guidance*, establishing an automated guide for the process by means of computerised tools. *(5) Automate execution support*, facilitating process enactment.

Static process modelling tools and techniques, such as RADs [3, 4], IDEF [5, 6], Workflow [7], etc. are used to capture processes to help in their understanding and to highlight the important aspects of them, such as organisational goals, analysis of activities, protocols, etc. However, on the other hand, such techniques do not generally capture the dynamic aspects (e.g. time, process instances, etc.) of these processes and therefore cannot fully predict the results of any proposed changes which might be applied to them, whereas simulation models aim to do exactly this [8, 9]. Thus, if we consider interfacing an existing process model representations with a suitable simulation tool, this might lead us to a position where a more complete and accurate approach to the analysis and management of processes is possible.

In this paper we present some results arising from the development of a prototype tool to support such an approach. The tool is concerned with the mapping of a process captured using Role Activity Diagrams (RADs) to the input to a Discrete Event (DE) simulator (fig. 1). The aim is to facilitate the generation of Discrete Event Simulation (DES) from process models. Figure 1 illustrates this approach. Here a process is considered as a group of related tasks which are performed by people and an IT system interacting together to achieve the goals of an organisation. First the process is analysed and captured in RADs, in terms of its activities, roles and their interactions, and decisions. Once the process is represented using this technique, a DES model is generated by a mapping to the elements of a DES software package. This DES model allowing for a more dynamic study of the process by means of "what if" questions. The key element in our work is to establish, as a first step, the differences and common features between the concept domain in RADs and DES Modelling. Identi-

[*]AIMG would like to thank CONACyT (México) for its financial support to carry out her PhD studies

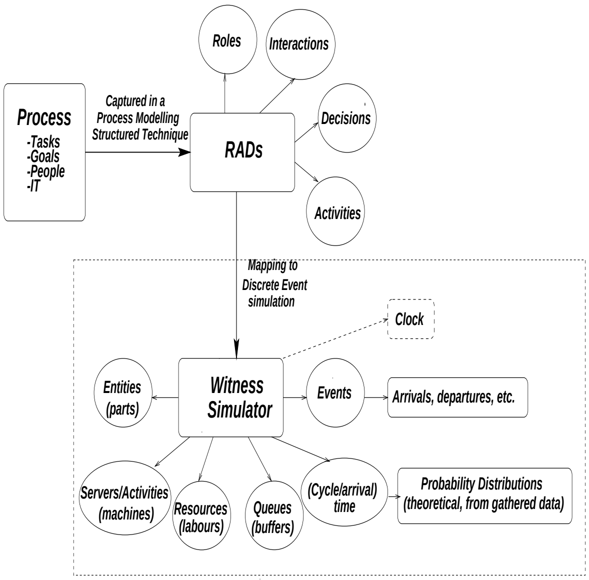

Figure 1: Process, Role Activity Diagrams and Discrete Event Simulation Models in terms of their main concepts.

fying the issues concerning the mapping from one kind of model to the other with the objective of facilitating the use of these kinds of tools. Thus, it is hoped avoiding extra effort in building the simulation models from scratch, and allowing the evaluation of changes to be influenced by the results of a simulation study. This, in turn encourages the reuse of previously built process models and promotes a common ground for the communication and understanding of the basic process aspects: that is the functional, organisational, behavioural and informational. The rationale for the use of RADs is its facility of mapping to different paradigms without undue difficulty. The RADs models can be easily enacted by mapping to the programming language PML [10] and in this paper the mapping to DES will be illustrated.

The approach is demonstrated using a case study carried out with the UK Manchester Community Health Trust. In particular with the information process concerned with the follow up of children's health and development. The corresponding process model is very complex. It contains many roles with many different types of interactions between them. Some of the interactions produce dependencies between activities. Thus, if there are insufficient resources to carry out all these activities in a given time, bottlenecks due to lack of resources and as a result of bad coordination in timing can be a consequence. It also has different cycles, replanning and rework possibilities, where timing is important to avoid any of the problems previously specified. In our approach the concern is not specific to this domain of application. It is clearly a general approach that could benefit any kind of complex process, either software, business or any type of organisational process.

The paper starts by giving a brief description of the case study followed by an introduction to Role Activity Diagrams and Discrete Event Simulation, in particular the WITNESS simulation software. Once this basis has been established, a brief description of the tool and the proposed mapping from RADs to DESs is given. This is followed by an example demonstrating the tool use and finally some conclusions are drawn.

2 Case Study

The case study was developed in the health sector with the Manchester Community Health Trust (MCHT). The aim was to model the nursing process in order to introduce process modelling to the MCHT and to apply our approach to the models. In particular, the process modelled is related to child health and development. This is the work carried out by Health Visitors[1] (HV). The model concentrates on the information-intensive process carried out by them (Child Health Information Process (CHIP)).

A rich picture of the process is presented in fig. 2. A brief description of the process is as follows. There are three main parties involved, the Hospital Trust, the MCHT management and HVs. Most of the roles involved in the process are owned by the HVs and the rest mainly by the MCHT, in particular by their administrative offices at Mauldeth House in Manchester.

Eight weeks after birth, the health and development of every child is monitored by a Health Visitor. Once the child is born the hospital sends the notification of birth and the hospital discharge letter to the MCHT administrative offices (Mauldeth House - MH). MH processes the information and inputs it into the Child Health System (CHS). Once this has been done, they will send the notification and discharge letter to the corresponding health visitor (for the particular child). Then the HV will arrange the first visit with the guardian/legal parents of the child, to check the child's and the mother's health. This is known as the Primary Visit. During this visit the HV will gather the correct and complete details about the child and his family (correct name, family's health history, etc.) The HV will also get the immunisation intent form filled and signed by the child's guardian/legal parents and with this (if approved) the immunisation process will get initialised. The primary visit also triggers the developmental assessment checks and in this way the process continues until the child

[1] Health Visitors in the U.K. are nurses specialising in the health and development of children under 5.

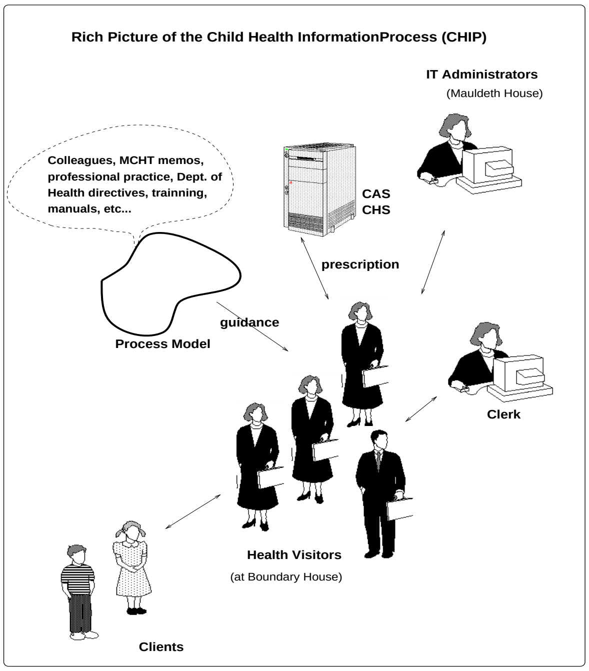

Figure 2: Rich Picture of the Child Health Information Process (CHIP)

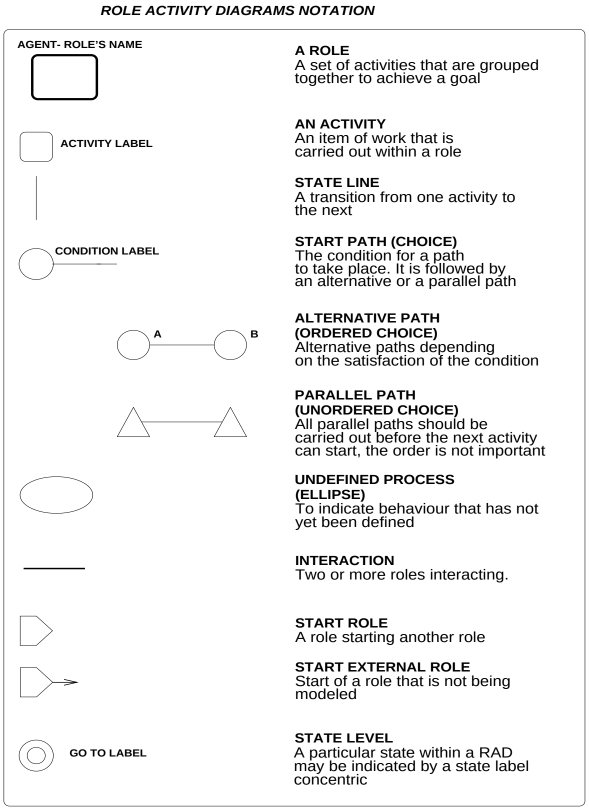

Figure 3: RAD's Main Graphical Notation and Elements

moves to another area, or another health trust or starts school.

The roles and interactions of the process were captured in RADs. A brief introduction to this technique is given in the next section. The role interactions diagram (fig. 6) illustrates the complexity of the process. Here the child is being used as the unifying element of the process. The model contains 22 roles that involve 4 subprocesses: Introducing Family/Child Data into the Trust System, Administering and Removing Data from the Trust, Administering Clinics and Running Clinics. Thus, this is a very complex and interesting model to simulate and analyse to determine how the metrics obtained from a simulation study could help in process improvement. We will use later the first 4 roles of the process, which we will call "the birth notification process" (fig. 7), to demonstrate and evaluate the tool.

2.1 Role Activity Diagrams

The methodology used for the study and capture of the process is the "Role Model" [11]. Which is composed of the following elements: a)*System Model*, a structural view of interactions between agents, b)*Goal Model*, specialises interactions by means of the goals that they can achieve, and c)*Method Model*, this shows how the goals of the interactive behaviour can be achieved, mapping the goal model into Role Activity Diagrams (RADs), and decomposing it into activities and interactions [11].

Part of the "Role Model" methodology is therefore the production of RADs. RADs is a structured technique for modelling processes, it focuses on roles, their interactions and their constituents activities, as well as the connection with external events [4]. This technique allows for a comprehensive approach to representing most of the features of a process (goals, roles, decisions, etc.) [12]. We concentrate on the use of RADs in this study, rather than the other elements of generic Role Modelling.

The RAD's graphical notation is illustrated in fig. 3 and a brief description of its elements is:

- **Role:** A set of activities that are grouped together to achieve a particular goal. In the graphical notation a role is represented by a rectangle with a label at the top and containing its activities. The label is compose of the *agent* and *the role's name.* Where the *agent* is a human or machine performing the role.
- **Interactions:** The representation of communication and coordination between roles.
- **Activity:** The items of work that are carried out within a role.

- **Start paths:** The condition for the paths to take place.
- **Alternative paths:** this are chosen depending on the satisfaction of some condition.
- **Parallel paths:** Paths, which are all carried out in parallel, the order is undefined, but before continuing with the next activity all the threads of activities on the paths must have terminated.
- **Undefined Process:** To indicate behaviour that has not yet been defined.
- **Start Role:** To show a role starting another role.
- **Start External Role:** To start a role that is not being modelled.

2.2 Evaluation of the CHIP Model

The evaluation of the model was made in terms of its qualitative aspects, its usefulness as a means for communication and for the establishment of the process baseline. It was carried out with members of the MCHT involved in the APPUS[2] project, by means of semi-structured interviews and a workshop with the application of a questionnaire. The process, as captured, is illustrated in the model in fig. 6. Whilst modelling the process some process problems emerged, in particular communication problems. The CHIP model motivated the communication and analysis of the process activities between the parties involved. Questions arose such as: Why do we do things this way? Is it possible to minimise the number of activities in the process? Which activities can be eliminated, if any?, etc. The model was found to be a very useful aid to such discussions. The notation used could be followed without any difficulty following a brief explanation. In particular the CHIP model helped the management staff to obtain a better overview of all the work involved in health visiting in addition to their familiar professional activities. For the HVs and IT managers, it helped them to gain a better understanding of how the processes fitted together. It brought up some inconsistencies in the way in which they carried out some activities and also identified, the type of information that could be provided by the IT system to the HVs.

Most of the professional work of health visiting is documented in the form of protocols and legal documents. However it is mainly the professional work which is documented. Thus, significantly, making the CHIP model, a rich source of documentation of the total information process.

[2]APPUS (Assisting Process Performance and Utilisation through Simulation), EPSRC Research Grant GR/L/15043

The MCHT staff found the model very useful in many ways, such as in documenting their processes in more detail and in finding inconsistencies in them. However, they felt the need for a complementary tool such as simulation, which will support their decision making in the management of their processes and in facilitating their analysis. Some of the results and comments from the workshop [13] which compared separate Process and Simulation Models of the MCHT were: "Most felt that process modelling was effective for communicating and that the documented process (model) can be accessible to more people", "Most found simulation easier to understand because of its Visual representation (process animation) and data collection results (more intuitive)", "Most felt that both techniques complement each other", "Simulation gives a good overview of things happening simultaneously", "Process models are clear and give extra detail", "Simplicity plus detail and focus". Thus, the need for a tool which supports both, the process modelling activity and simulation modelling was clearly evident. A tool which automatically generates simulation models from process models facilitating the mapping from one to the other, clearly might be a useful tool in such scenario. However, training for experimentation and statistical analysis will still obviously be required.

3 Discrete Event Simulation

DES is the most adequate simulation paradigm for this kind of modelling. This can be defined as the process of modelling and studying a given system, and predicting its future dynamic behaviour from its given current state.

The basic elements of DES models are entities which have some properties or attributes. The state of the system is a collection of attributes or state variables that represent the entities of the system at a given time. At these times (different discrete points in time) events occur. The system state may change when an event happens. Any such event starts an activity that occupies a length of time, during which the entities of the system perform some operations. The entities, attributes, events, activities (servers) and the relationships between them are defined in the model of the system [14].

Simulation is a very useful technique, "the ability to visualise how a process would behave, measure its performance, and try "what ifs" in a computer model makes computer simulation an invaluable tool for decision making" [15]. It allows for the performance of studies where some metrics can be gather. Bottlenecks can be analysed through the observation of the utilisation of resources, cycle times, processing capabilities, waiting times, allocation of resources, coordination, etc. [9, 16].

For the implementation of the prototype tool, simulation software satisfying the requirements for visual representation and animation, ease to use, statistics gathering, etc. was considered. The WITNESS simulator was the software selected to be used in the prototype tool. A brief introduction to its modelling technique and main features is presented next.

3.1 WITNESS

WITNESS[3] is a discrete event, visual interactive simulator. It allows the visual representation and animation of the elements of the system arguably making it easier to analyse their behaviour [17, 18].

WITNESS models are built in three steps: *DEFINE* the elements to be in the model(their names, type and quantity); *DISPLAY* the elements (its not compulsory to display each one, just the necessary elements to understand the simulation) to visualise the model and its animation; *DETAIL* the elements. Here the behaviour of the model is defined in terms of details such as: average service (cycle times) capacities, waiting times, arrival time, rules for the behaviour of the system, etc.

WITNESS has 22 pre-defined elements [17, 18] and these can be divided in 2 different types: PHYSICAL and LOGICAL Elements:

1. **PHYSICAL ELEMENTS.** (These are displayed as dynamic icons and represent tangible entities in the real life situation being studied): **Parts (*entities*):** Flows through the model. i,e. products, project progressing, telephone calls, people, components, etc.; **Buffers (*queues*):** In which parts can be held waiting to be processed; **Machines (*servers, activities*):** Anything that takes parts from somewhere and processes them. There are six different kinds of machines in WITNESS: **Single** (processes one part at a time), **Production** (takes in a single part and outputs a number of parts), **Assembly** (takes in a number of parts and outputs a single part), **Batch**, **General**, **Multi-station**; **Labour (*resources*):** This is a resource (for example tools or human operators) which may be required by other elements for processing; **Conveyors**, **Vehicles**, **Tracks**, **Processors**, **Fluids**, **Tanks**, **Pipes**, **Modules**.
2. **LOGICAL ELEMENTS.** (These represent the data and reporting aspects of the model, that is the control of the model.): **Attributes**, **Variables**; **Distributions:** Used to build variability; **Functions:** To build intelligence into the logic of the model, **Time series, Histograms, Pie Charts:** Present results on the screen as graphs (**REPORTING** elements), **Shifts**, **Files**, **Part Files**.

WITNESS uses input and output rules to specify the general behaviour of the process, that is, to specify the flow of entities through the system.

With all these elements outlined the translation process from RADs to the elements of a DE model, in particular in WITNESS, is now described in more detail in the next section.

4 The tool

In the prototype, it is assumed that process models have been captured in RADs. From them, the DES models are generated for the WITNESS simulator which, as just described, supports modelling in terms of parts (entities), machines (servers), labours (resources), buffers (queues), etc [17]. In particular, the basic process information is captured using the Process Modeller Workcentre[4] (PMW) in terms of RADs. The first mapping from its elements (activities, roles, etc.) to the WITNESS simulation elements (machines, buffers, parts, etc.) is performed as illustrated in fig. 1. Because RADs are not time-based, the technique does not identify the dynamic properties or attributes of the activities that are needed for the simulation. Thus, missing information (some simulation attributes such as cycle times, arrival times, number of instances for a particular element, etc.) for the experimentation is obtained directly via the user interface. With this information the mapping and the final simulation model can be generated in the notation of the WITNESS Input Command Language. The model is generated to maintain consistency with the way in which the process was captured in RADs. The model is automatically loaded into the WITNESS environment and then very basic simulation (and animation) of the process can be performed.

The prototype thus translates a static process model into a dynamic simulation model that can be used for calculating service times, analysing bottlenecks, etc. Hence the user can then plan the simulation analysis that is required in a particular context.

The overall structure of the system (fig. 4) has the following elements: 1. **Basic Mapping.** A first mapping of the existing attributes in the RAD model to the WITNESS simulator code. 2. **User Interface.** The

[3]WITNESS AT&T Istel is a discrete event visual interactive simulation tool, commercialized by Lanner Group

[4]PMW, developed by the Informatics Process Group of the University of Manchester

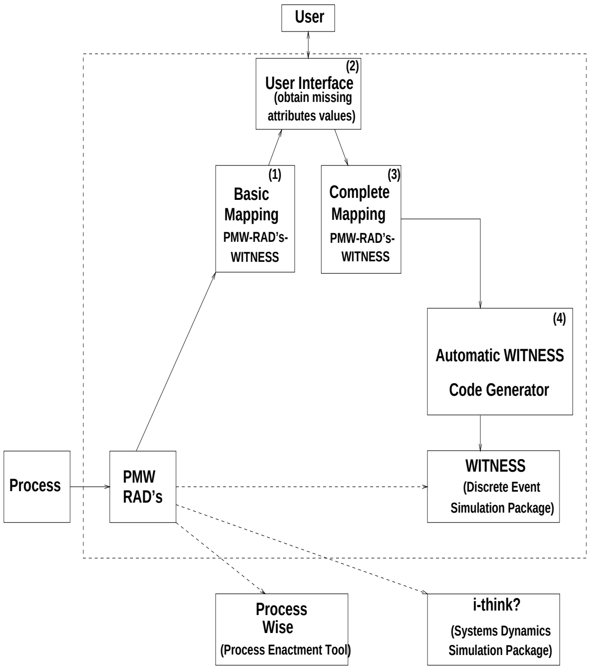

Figure 4: General Design of the Automatic Simulation Model Generator

different models do not map directly from one to the other (there is a lack of information on the RAD model) and a way is needed for providing the extra attributes for the simulation model. The missing information in the RAD model is directly input so that there is a more complete WITNESS model at the time of the second mapping. 3. **Complete Mapping.** After adding the missing information we have a complete mapping and hence a complete model. 4. **Automatic WITNESS Code Generator.** This is the final system.

From the analysis of RADs and DES main concepts, a basic mapping between the elements of both models was established. Essentially the elements of the RAD model: roles, activities, decisions and interactions, were mapped to entities (parts), resources (labour), servers (machines), and queues (buffers) in the simulation model.

4.1 Mapping

An important element in RADs are the activities. These represent the elements of work in a process, and are represented by servers in the simulation, defining a queue for each one of them. Roles are performed by agents (often humans) in a process model, while on the other hand resources need to be allocated to a server to be able to provide the service in the simulation. Interactions and state lines in a RAD provide the information which define the rules concerning the flow of entities through the system in the simulation. However although this mapping appears to be smooth, there are still problems in gathering the information to establish the behaviour of the system. Information such as that related to the cycle time for the servers as well as for the entities inter arrival-times. In the case of decisions, the statistical information needed to define the flow of the entity according to the condition is not available in the RAD and therefore the user has to introduce it directly. The structure of the mapping that was finally used is shown in fig. 5. More details of this mapping are now outlined. Although the mapping is from RADs to WITNESS (DESs), it is simpler to explain it in terms of the WITNESS elements.

RAD mapping to WITNESS

RAD element →	WITNESS element
ROLES	1 PART
	N LABOURs
ACTIVITY	1 BUFFER
(when it interacts with another role and the activities continue in the two roles)	1 Production MACHINE
(when there is an interaction coming from another Role and also a link from a previous activity)	1 Assembly MACHINE
(in any other condition)	1 Single MACHINE
THE START OF ALTERNATIVE OR PARALLEL PATHS (CHOICE)	1 BUFFER
(when followed by alterantive paths)	1 Single MACHINE
(when followed by parallel paths)	1 Production MACHINE
THE END OF PATHS (END CHOICE)	
(when terminating alterantive paths)	1 BUFFER
	1 Single MACHINE
(when terminating parallel paths)	N BUFFERs
	1 Assembly MACHINE
ALTERNATIVE PATH (ORDERED CHOICE)	The same as activities
PARALLEL PATH (UNORDERED CHOICE)	The same as activities
START ROLE	The same as activities
START EXTERNAL ROLE	The same as activities

Terminology
General Simulation -> WITNESS

Entity	-> Part
Queue	-> Buffer
Server Activity	-> Machine
Resource	-> Labour

Figure 5: Mapping from RADs to WITNESS elements.

PARTS: PMW-RADs' representation has an attribute for the *Roles* to identify which of them will trigger the simulation. This attribute is called "*first role*" and the value can be *false* or *true*. When generating the parts for the simulation we take the *role* of the RAD model with the former attribute equal to "*true*". From this role we define the **part** (entity) that will move through the system.

LABOURS: An important aspect of the mapping is the fact that, when modelling organisational activity, most of the activities are carried out by people and it is important to represent this in the simulation. Thus, the concept of the Worker Interference Model (WIM) [19] was used as the basis for the generated model. Here, in order for a task to be perform, there is the need for a worker (labour in WITNESS) to be present. Hence, for each *role* of the RAD we generate a number of **labours** which will perform the *activities* of that *role* in the simulation. The particular number of **labours** assigned is given by the user by a very simple user interface.

MACHINES: As described before, there are six different kinds of machines in WITNESS: Single, Production, Assembly, Batch, General and Multi-station. The type of **machines** used in the mapping are **single** (processes one activity at a time), **production** (takes in a single entity and outputs a number of entities) and **assembly** (takes in a number of entities and outputs only one.) **Production** machines are used when after a simple activity there is a need to perform two or more different threads of activities, therefore one or more copies of the entity need to be produced. The use of **Assembly** machines is in the cases when, in carrying out an activity, there is a need to wait for the results of the processing of two or more previous different activities. Basically a **single machine** is generated for all the *activities* on the RAD model, the *alternative paths*, *parallel paths*, and *undefined activities*, *start role*, *start external role* and for the *start of alternative paths* (the condition for the paths to take place).

In the particular case when a *start path* is followed by *parallel paths*, the *start path* is mapped to a **production machine**, hence the **part** feed into the **production machine** will be multiplied by the number of parallel threads of activities that need to be performed. In PMW-RADs the *end paths* are the elements that mark the end of the set of activities carried out within a *start paths*. When the *end path* is the terminator for the activities coming from *parallel activities*, then the *end path* is mapped to an **assembly machine**. This means that this part of the process cannot continue until all the activities from the parallel paths have been completed. The time taken will be the time of the path that takes the longest to finish its activities.

BUFFERS: **Buffers(queues) are waiting or queueing elements**, therefore a **buffer** is generated for each **single** and **production machine** and **N buffers** for the **assembly machines**, according to the number of *parallel activities* that have to complete their work before the next *activity* can be carried out. **Buffers** are an important element of simulation, as they allow the monitoring of waiting times, queues, etc.

An interesting question is why is it necessary to introduce **buffers** to all the **machines** if some may not need them? A **buffer** is defined for each **machine**, because in a generic tool, based on RADs, such as in this system, it is not possible to know in advance which activities will need them. The time each activity takes is non-deterministic, indeed it may be instantaneous. Nevertheless it is necessary to specify some time (probability distribution) for the completeness of the model. This redundancy using dummy activities does not cause problems, as long as their times are very small compared with the relevant activities. It is also worth remarking that in this way the structure of the original model, is retained making it easier for the domain experts in the organisations to understand and become familiarised with the simulation model.

INPUT and OUTPUT RULES: **Input** and **output rules** are those that determine the flow of entities through the system. In the RAD model this is given by the state lines (links) between the *activities* and from the *interactions* between *roles*.

When *alternative paths* exist, RADs does not explicitly capture information about the probabilities of the entity flowing to any possible path. Hence, some assumptions are necessary. In this case, very simple **rules** are used to determine the behaviour of the process. At the moment **Output Rules** with a given percentage of possibility of taking any of the possible paths is used, but clearly any other type of rule or distribution could be utilised.

PROBABILITY DISTRIBUTIONS: At the moment a theoretical probability distribution is used: a **Negative Exponential** with different service times to define the **machines' cycle time** and the mean of inter-arrival time for the **parts**. However this can be easily modified to introduce any theoretical or empirical probability distribution.

4.2 Evaluation

The evaluation of the prototype was performed initially, with simple models to check the consistency of the mapping and the simulation model. Then further evaluation was made using more complicated models such as CHIP, the case study, outlined here.

The prototype performed an adequate mapping gathering the missing information directly from the user, as previously specified. One important feature of the mapping in this prototype, is the fact that the simulation model retains the same structure which was established in the process model (fig. 8). Thus, facilitating the understanding of the simulation model by those already familiar with the process model in RADs. This in turn, avoids making the model more complicated by having to make abstractions of a different kind. Thus allowing the modeller to concentrate on the experimentation and facilitating the communication process.

This prototype does not automate the experimentation process. This is very specific to the application domain and the process owner. The manager or process owner needs to plan a proper simulation study according to the specific domain and activities for investigation (i.e. for process improvement or redesign).

The mapped simulation model reproduces the behaviour of the process model captured using RADs. When running it, the simulation advances through the activities (events) in a compressed time, animating the RAD process model. This gives a general overview of the important aspects of the process which are occurring simultaneously. In contrast with the RAD process model, it is possible to view multiple instances of the process acting together, and thus allowing simulation studies which will reflect more closely the real process, to be performed.

Some experiments were made with some of the roles of the CHIP model. In particular with what was termed earlier the "Birth Notification Process" which is illustrated in fig. 7. The experiments investigated timing, waiting times, processing time, bottlenecks and coordination of subprocesses.

When interacting with people, in order to model a process, then generally the staff concerned can identify the activities involve in their processes and therefore the process can be captured in terms of what activities and interactions are involved. However when simple data related to the timing of the process is required, it is not as easy. In order to have a reliable and real simulation model, the data should reflect the actual process as closely as possible. The simulation model was generated with approximate times given by the process owners. Thus, the analysis of the experiments was not exhaustive in statistical terms. However a global analysis of the process with respect to its critical activities was possible.

The experiment concentrated on the analysis of bottlenecks due to lack of coordination in the interactions and to the finite number of resources available to perform the activities. In particular we can observe the queueing effects in the "Primary" role of the model (fig. 7), which is owned by the HVs. Suppose the "Birth Notifying" process performed by administration is very efficient and takes little time to interact with HVs. On the other hand if the "Child Birth Notification" and the "Handling Discharge Latter" processes together take longer than the first one, then there is a queue in the activity (Primary, 3) that interacts with the outputs of both subprocesses. This activity (Receive child's Discharge Letter from MH) is an assembly activity, this means that it cannot continue until both, the previous activity and the interactions have taken place. It is only then that the next activity can be performed. Another interesting point is concerned with the utilisation of resources. For example, in the "Primary" role, another significant behaviour occurs. Originally, as in the real process, the simulation model was created with only two HVs to carry out the activities encapsulated within this role. When the simulation is performed a large queue is observed forming at the beginning of this subprocess (Primary) and in activity 3 of this same role (fig. 8). In this case this is because the number of activities to be carried out by the two HVs are too numerous and there are only two HVs. Thus, the work is accumulating at the beginning of the subprocess (the buffer of the first activity). The queue in activity 3 is a consequence of the first one. This behaviour can be observed in the graph of fig. 8, where the bottom curve corresponds to the queue at the beginning of the "Primary" role and the top to the activity 3 of the same role. As we increase the number of resources, HVs, the problem disappears. This raises a problem that can be addressed either, by restructuring the process such that, the number of activities or the timing to perform the activities is decreased, or by increasing the number of resources. In either case, the same type of simulation study of the new process can then be performed to ensure improved process management.

5 Conclusions

Even though both approaches, process modelling and simulation, are concerned with similar issues, the type of information that is gathered for the analysis and representation of their models is quite different. RADs do not capture some of the basic information needed for statistical analysis of a process because it is not time based. One advantage of combining process capture and simulation, is to increase the probability of the automatic generation of a simulation model from a process model whilst retaining the structure of the initial process model. This provides the capability for initially generating a RAD model of an abstract view of the process and subsequently adding refinement of the model as more information about the process is being defined or obtained [20]. Thus introducing more but appropriate complexity to the process model. Once this detail hierarchy has been obtained it can be used to examine a large model in terms of specific simulations of behaviour at chosen parts of the process. The RAD notation allowing for the development of an easy to understand model and the simulation for the detail inspection of specific parts of the process. The combination of the two allowing for a more structured, and considerably, cheaper approach to the simulation of very large processes. Rather than produce expensive single level simulations of total processes we perceive the joint approach, as illustrated by our prototype, allows for a refinement approach to systems simulation. Essentially this combines the advantages of abstract models, using the RAD technique, with the advantage of simulation of very concrete models to investigate specific process bottlenecks.

From the experience obtained from the development of the tool, we can identify the need to modify the RADs' representation in order to introduce all the necessary dynamic attributes at the time of capturing the process for the generation of a complete simulation model. At the moment we are working on these extensions. The key is clearly to introduce the extra semantics in such a way as to not lose the clarity currently associated with the RAD notation.

References

[1] Snowdon R.A., *Overview of Process Modeling*, URL: http://www.cs.man.ac.uk/ipg/Docs/pmover.html, consulted in September 1997.

[2] Curtis B, Kellner M. I. and Over J., *Process Modeling*, 1992, Communications of the ACM, *Volume 35, Number 9, Pages 75-90.*

[3] Holt A. W., Ramsey H. R. and Grimes J. D. *Coordination System Technology as the Basis for a Programming Environment*, 1983, Electrical Communication, *Volume 57, Number 4. Pages 307-314.*

[4] Ould, M. A., *Business Processes: Modelling and Analysis for Reengineering and Improvement*, John Wiley and Sons Inc., 1995.

[5] *IDEF Methods*, URL: http://www.idef.com/.

[6] Benjamin P. C., Mayer R. J., Blinn T. M., *Intelligent Support for Simulation Model Design from IDEF3 Descriptions*, October 1992, Proceedings of the IDEF Users Group Conference.

[7] *New Tools for New Times: The Workflow Paradigm*, edited by Layna Fischer, Future Strategies, 1995, Second edition.

[8] Pegden, C. D., *Simulation: A look to the future*, 1997, 11th European Simulation Multiconference (ESM'97), *Pages K17- K21.*

[9] Gladwin B. and Tumay K., *Modelling business processes with simulation tools*, 1994, Proceedings of the 1994 Winter Simulation Conference, *Pages 114-121.*

[10] Warboys, B.C., *The IPSE2.5 Project: Process Modelling as the Basis for a Support Environment*, May 1989, In Proceedings of the First International Conference on System Development Environment and Factories, Berlin.

[11] Informatics Process Group, University of Manchester, edited by Brian Warboys, *Business Information Systems : a Process Approach*, 1998, in preparation.

[12] Miers, D. in *Business processes reengineering*, edited by Coulson-Thomas Colin, kogan Page Limited, 1996.

[13] Kay S., Hardiker N., Aldridge S., Warboys B., Robertson I., Martinez-Garcia A., Deneath M.,Pitt M., *Process Modelling and Simulation for Managing Clinical Care in the Community*, 1998, in revision for the 1998 AMIA Annual Fall Symposium.

[14] Law, A. M. and Kelton, W. D., *Simulation Modeling and Analysis*, McGraw-Hill, Inc. Second edition, 1991.

[15] Tumay K., *Business Process Simulation*, 1996, Proceedings of the 1996 Winter Simulation Conference, *Pages 93-98.*

[16] Swami A.. *Building the Business Using Process Simulation.*, 1995, Proceedings of the 1995 Winter Simulation Conference, *Pages 1081-1086.*

[17] AT&T ISTEL, *WITNESS 6.0 user manual*,1994.

[18] Thompson, W. B., *Introduction to WITNESS and linking to process mapping tools*, 1995, Proceedings of the 1995 Winter Simulation Conference, *Pages 462-466.*

[19] Buss, A. H., *Modelling with Event Graphs*, 1996, Proceedings of the 1996 Winter Simulation Conference, *Pages 153-160.*

[20] Bruce Silver Associates, *BPR Tools Add Power to Workflow Automation*, Industry Insider Report URL: http://www.metasoftware.com/whiteppr.html, consulted in September 1997.

Author Biographies

Ana I. Martinez is currently a PhD student at the University of Manchester. Her research project is concerned with the use of simulation as a tool to understand better the dynamics of process models. She received her MSc degree from the same university in 1996 working on the study of discrete-event simulation modelling for rapid prototyping. She has participated in workshops on process models and simulation in the health sector. She is particularly interested in the application of both process modelling and simulation applied to the re-engineering of organizational processes.

Brian Warboys has been the Professor of Software Engineering at the University of Manchester for the past twelve years. Previously, since 1963, he worked for ICL, the UK's largest computer manufacturer, and was Chief Designer of the company's mainframe operating system VME during its formative years in the 1970's. This experience led to an ongoing research interest in the problems of designing very large software systems. Focus on the development process for such systems and indeed, business processes more generally, proceeds through the work of the Informatics Process Group (IPG). The IPG was formed in 1979 and is particularly concerned with the general problems of realigning systems to co-ordinate and to integrate the wide variety of activity within the organisation, both human and machine, both co-present and distributed.

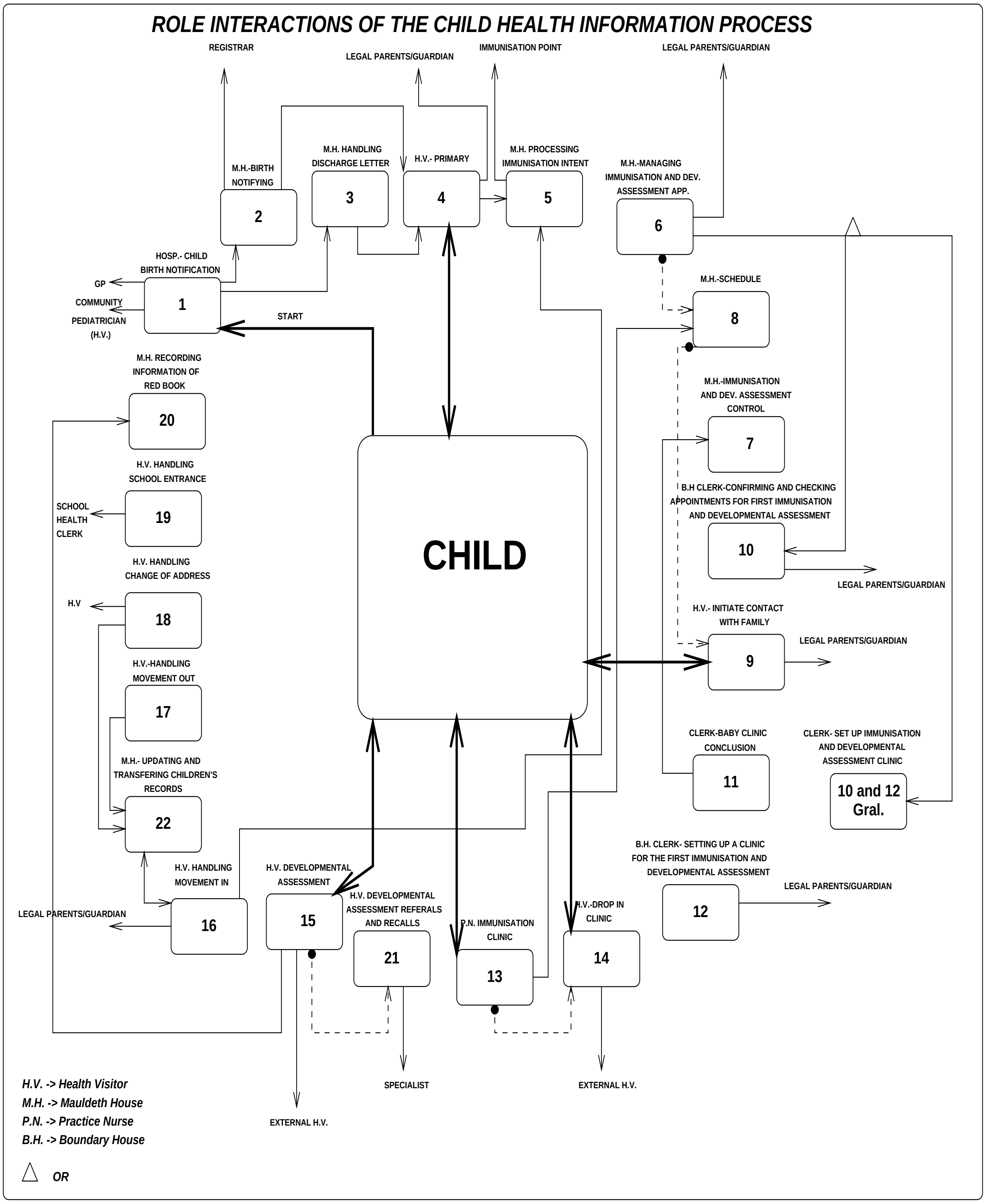

Figure 6: Role Interaction Model of the Child Health Information Process.

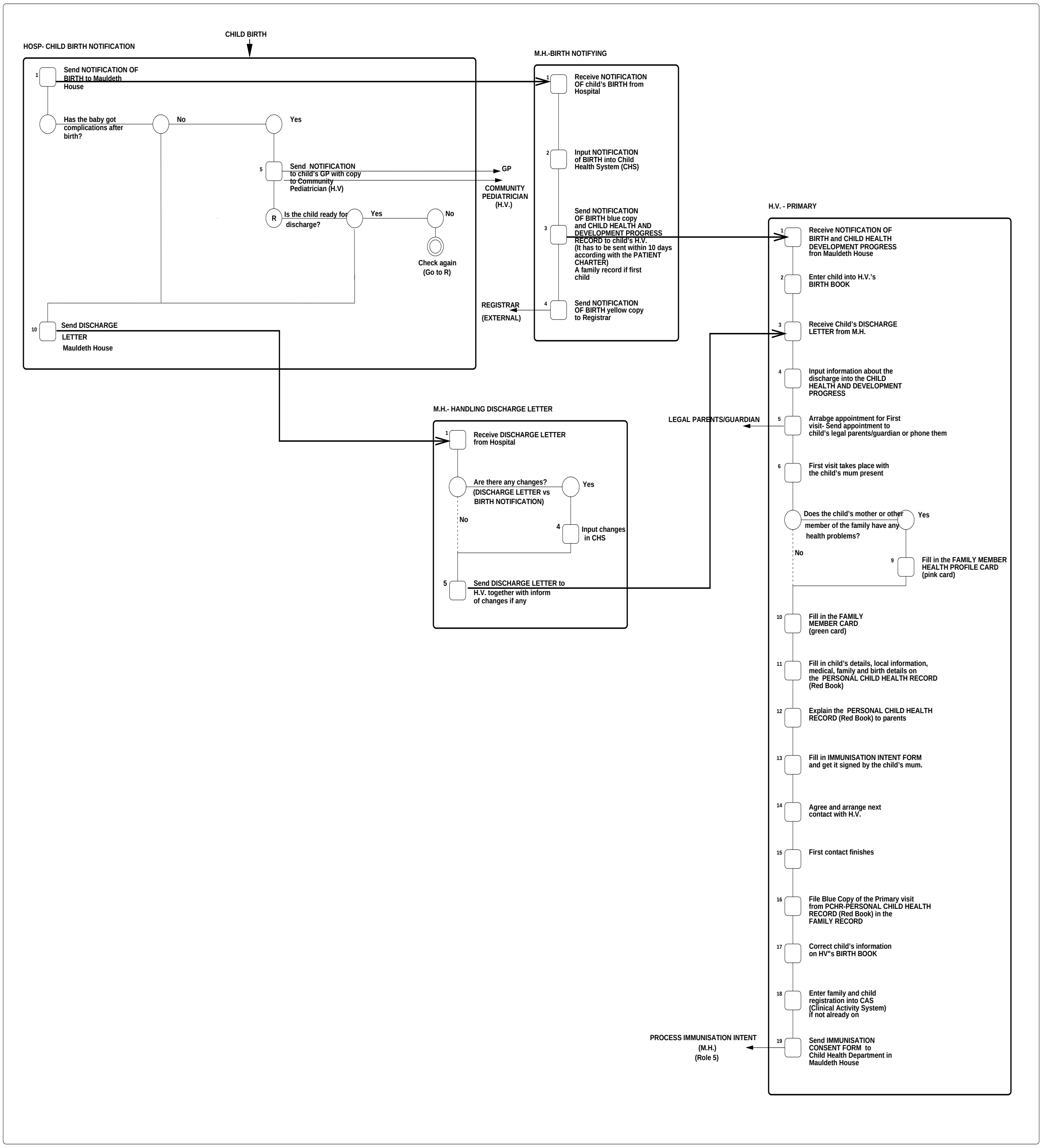

Figure 7: Some Roles of the Child Health Information Process (the birth notification process).

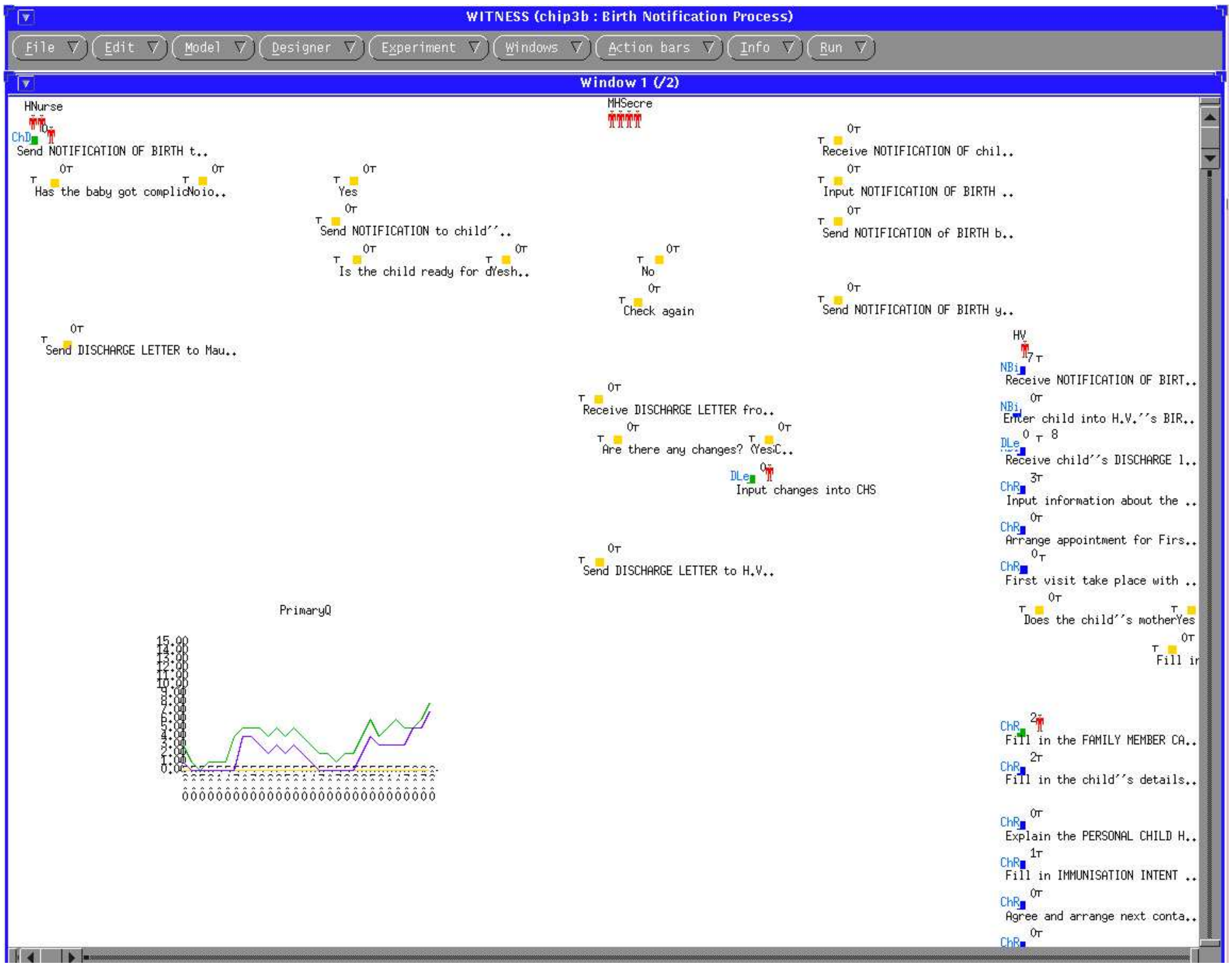

Figure 8: Child Notification Process. A Subprocess of CHIP simulated in Witness