FROM PROCESS MODELLING TO ENACTMENT AND SIMULATION

Fernando Rojas and Ana I. Martínez *
Informatics Process Group
Department of Computer Science
University of Manchester
Oxford Road, Manchester M13 9PL, UK
e-mail: {rojasinf,martinea}@cs.man.ac.uk

ABSTRACT

The usefulness of simulation and enactment in business process modelling is explored by means of studying a simple process. Two roles with interactions (contacts), rework and choices collaborate in the production of a document (loan, policies, etc.) From its Role Activity Diagram (RADs), we construct models for Discrete Event (DE) and System Dynamics (SD) simulation and Process Enactment (PE) in a computer system. Our work emphasises the reuse of process models already existing in organisations by constructing simulation models from their static representations (RADs) (keeping the process baseline unchanged). A brief description of the process elements and how they can be mapped or represented in DE, SD and PE is provided.

INTRODUCTION

Process simulation is a powerful tool that helps in making better decisions concerning the improvement and redesign of organisations through a processes reengineering initiative (Swami 1995). In many organisations an important effort has been given to mapping their processes using a variety of structured techniques. This, with the objective of obtaining a better understanding of their processes, so that (1) people can work together more effectively and provide information to the individual or group to perform the intended objectives, (2) evaluate the procedures to find those which are obsolete and redundant and, (3) use technology to support and monitor the processes. In this paper, the emphasis is that, a simulation study in an organisation, should encourage the reuse of a unified modelling framework in the description of their processes. This, promotes a better understanding and communication among the different agents (managers, engineering, technicians, secretaries) that constitute it.

The purpose of this paper is to present an study of some simulation paradigms to investigate process models. We also emphasises how to reuse the process modelling efforts previously carried out by an organisation in other contexts.

A description of the different characteristics of a a process with the objective of mapping them to a simulation model (discrete event, system dynamics) or enactment (automated process execution, driven by the agents) is introduced. This will allow us to assess the changes of conceptualisation from one domain to another and, on the other hand, to evaluate which provides the information required or is more appropriate to a specific type of process analysis. We do this through the analysis and evaluation of a simple process: Obtaining Signature Collaborator Agreement writer or OSCA model (Walker et al. 1996). This is a simple process that can occur in an organisation such as in a bank (the loan process) or in an insurance company (policy request). The process goal is that an agreement about the content of a document should be reached by two experts.

We assume that a company has process profiles of most of its operations and they were captured with the structured technique called Role Activity Diagrams (RADs) (OULD 1995). This technique maps or captures a process from tasks, goals, and people, to roles, activities and interactions. It provides much of the information required to characterise the main features of a process (activities, orders, decisions, roles, responsibilities for decisions, goals, etc.) In RADs, the roles carry out actions (activities) and make decisions of what to do and when according to the organisational goals. The OSCA model is represented in RADs and from them, we construct the corresponding simulation models. We develop a representation based on Discrete Event (DE) using the WITNESS simulator (Thompson 1995), for the System Dynamics (SE) approach using Ithink (Richmond et al. 1994) and for Process Enactment (PE) using ProcessWeb (Greenwood and Warboys 1996) together with the Process Modelling Language (PML).

The overall idea of the work is presented in Figure 1, where RADs act as a unified model of the process and

*AIM would like to thank CONACyT (Mèxico) for its financial support to carry out her PhD studies.

provides the basis for any type of dynamic study.

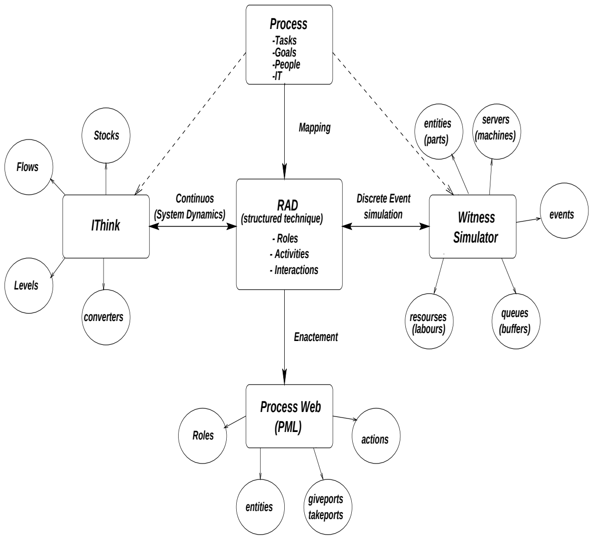

Figure 1: Representation of the objective of our work on process model simulation. RAD serves as a unified unbiased view of the process and from there we can map out to simulation, based on either the discrete or continuous approach and also to the enactment in a particular process support tool (ProcessWeb with PML), to see the actual process driven by the roles inputs and interactions.

We provide a brief evaluation and discussion on the sort of information obtained from the three approaches and the changes of conceptualisation required to map from RADs to DE, SD and PE. Finally, with simulation (keeping the process baseline unchanged), we want to evaluate some of the dynamic properties that could not be extracted from static models. For instance, we can obtain information on the overall process cycle time, the average time for processing an agreement, number of times an agreement is corrected, the utilisation of roles (time, people), activities (underutilisation, bottlenecks) among others.

MODEL IN RADS

RADs provide the main work-flow chain for the process without a specific domain application in mind. In our model it is being used only with the goal of capturing the main features of the process. This is a convenient approach as a starting guide for a simulation model construction, and enactment with or without data information support.

The OSCA model involves the coordination of two people (Agreement Writer (AW) and Collaborator (CO)). The former writes an agreement that is sent to the CO for approval, if the content of the agreement is correct the AW sends a copy of the final agreement to the CO, saves it in a file and the process is completed. On the other hand, if the agreement has to be modified because the CO considers that it is incomplete, the CO sends some comments, which are read and performed by the AW, modifying the document content and then sending it back again to the CO for revision. The iteration of correction is carried out until the agreement is complete.

The OSCA model is characterised by the following elements:

-Two roles: the Agreement Writer (AW) and the Collaborator (CO).

-The process goal is to produce an agreement document acceptable to the business rules with the collaboration of the two roles.

-Requires an interaction model between the two roles. We observe interaction when (a) the AW sends the agreements draft to the CO; (b) the CO sends back some comments on the agreement; (c) the AW makes the corrections and sends them back to the CO; (d) the CO accepts the content of the final agreement and finally, (e) the AW saves the accepted version and sends a copy to the CO.

After the description of the process structure, we proceed to map it to the Role Activity Diagrams showed in Figure 2.

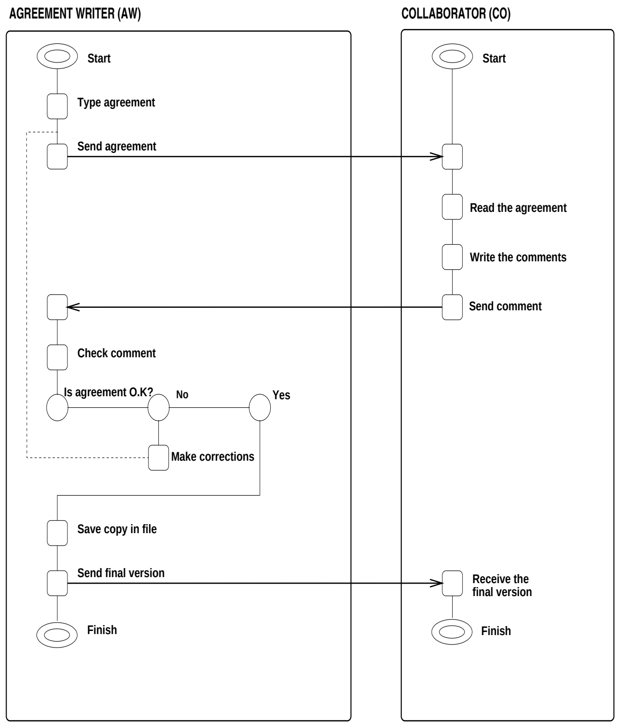

Figure 2: Role Activity Diagram for the OSCA model.

In the RAD representation, the OSCA model has two roles, (large boxes) where the square boxes inside represent activities or units of work that the role carries out with the purpose of fulfilling a process goal. The connection between activities shows the ordering of the process (state). RADs also represent choices as linked circles (*is the agreement ok ?*). The interactions are represented by horizontal lines connecting two roles. It has also the iteration that link two states in a role (dashed lines). RAD has more elements (OULD 1995) but for our purposes, we only described the necessary for understanding the RAD's representation of the OSCA model.

Clearly, the OSCA model has some of the features of complex processes (interactions, iteration, etc.) The points or states where the simulation can be useful are on the interaction and iteration parts. In the interactions the coordination between roles is a key element to avoid bottlenecks or white spaces. The iteration part (rework) could increase the process cycle times. We describe how to construct the simulation models (DE, SD) and the process enactment in ProcessWeb in the next section.

SIMULATION AND ENACTMENT

For the purpose of this work, we describe all the relevant elements required for mapping the OSCA to DE (entities, server, queues, events), SD (reservoirs, conveyors, flows, conveyor), or PE (roles, actions, entities, giveports, takeports).

ProcessWeb

The enactment is concerned with the implementation of a process in a computer system, with the purpose of supporting the process as it happens in practice. The implementation of OSCA is in ProcessWeb (Greenwood and Warboys 1996), a process support system that combines PML with the Web. The OSCA model design for this tool can be developed without difficulty from RADs. The concepts map smoothly from one domain to the other. ProcessWeb interfaces the roles using as a front end, web pages generated (HTML) for the specific state of each role during the process. The AW and CO are represented in a construct of PML called **Role**. Each role contains the **resources** used and the **actions** carried out. The resources describe ports for communication between roles and the entities which are the pieces of information used in the communications. The communication is performed with a resource of type *giveport* to send original or copies of entities to other roles and *takeport* to get information from other roles. The actions represent the activities of roles.

The entities manipulated and interchanged during the process are: *Agreement (Parts text: String and status: String)*, it contains two parts: the text of the agreement and the status, and *Comment (Parts Acceptable: Boolean and details: String)* . In a main role, we create instances of the roles AW and CO, the corresponding binding for the resources and the interactions for connecting the roles. The enacted process changes its state according the roles' inputs from an external agent (user) that is interacting with the process through Web pages generated by ProcessWeb. The program, then defines how the data introduced from the pages is used and which Web page needs to be updated to represent the current state. It is in this sense that enactment introduces the dynamic aspects of the process, but its purpose is limited to supporting it.

Discrete Event Simulation

The representation of the model has been done in WITNESS (Wit): a visual interactive simulator with capabilities for discrete event simulation, good animation and graphical display. A model is constructed following the steps: define (simulation elements required for the model: entities, server, queues), detail (input-output rules for defining the behaviour, service and arrival times and the logic to manipulate attributes) and display (graphical representation of the elements).

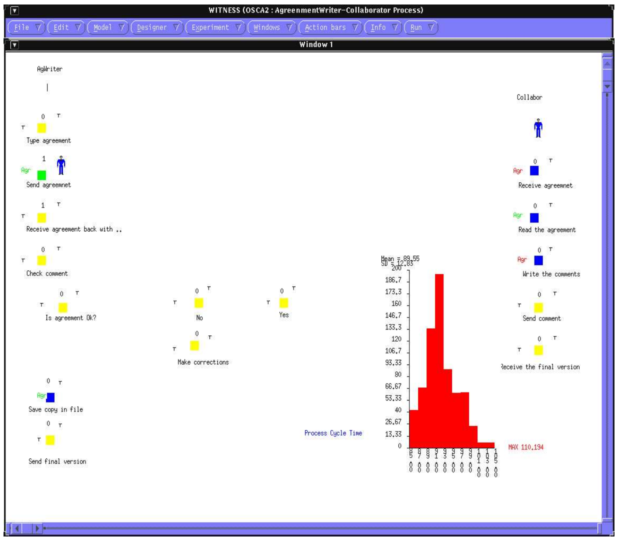

Figure 3: Discrete Event model for OSCA in WITNESS.

The entity in this case is the agreement, defined with two attributes to identify the rework process, and to stamp the total cycle time. The activities carried out by the AW and the CO are represented by a server (machines in Wit). Each machine has associated a queue (buffer in Wit) to gather statistics of bottlenecks. Each role has a labour associated, which is used to represent the resources. The rework process when the agreement has to be corrected, is represented as an event to move the entity to a specific element in the work-flow chain according to statistical input-output rules. Figure 3 shows a snapshot for the final representation of the OSCA model in WITNESS. We can extract information about the total cycle time of an agreement in the process. The accumulation of work in AW, due to the arrival of new agreements plus the ones to be corrected, can be observed and monitored with the average queue size of the buffer associated to the *send agreement* machine. Here is where the iteration part of the process (rework) can be measured.

System Dynamics

For SD modelling we use the Ithink tool (Richmond et al. 1994). It allows us to create simulation models using graphical elements (stocks, flows, conveyors, converters) linked together in a network to provide the functionality required.

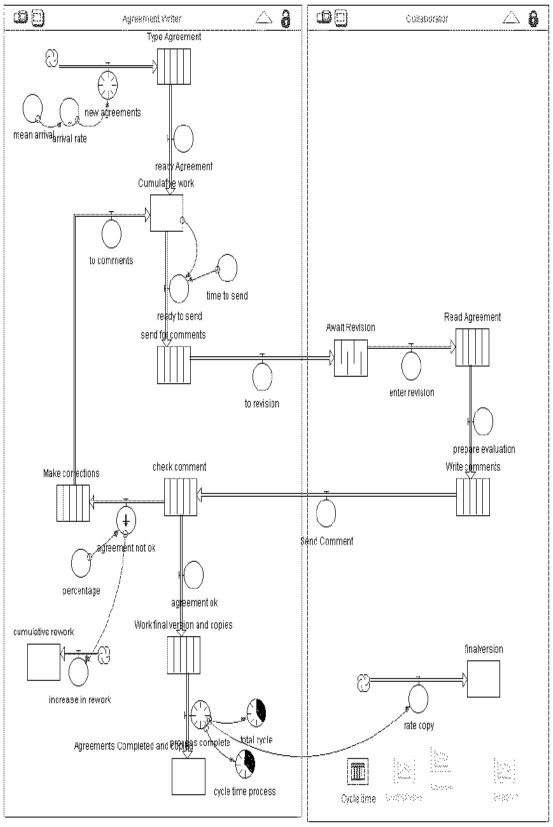

Figure 4: OSCA model represented in the system dynamics approach, using Ithink (a graphical simulation program).

For our model, Figure 4 shows the corresponding structure in the SD representation. The key activities are represented as conveyors (elements where the material stays inside for a certain amount of time, small boxes with vertical lines). The work accumulation (new agreements plus agreement for correction) are represented by a standard stock element (*Cumulative work*). The labour can be introduced in the immediate flow (*ready to send*) or in other flows in the chain, with the use of a support infrastructure for resources consumption. Because some material is reworked, the conveyor (*check comment*) has two outflows, one for the correct agreements and other for the ones that need modification. In the latter, we use in the flow (*agreement not ok*) a function called REWORK (*percentage*), that uses as an argument the percentage (converter) of material that has to be reworked. For the cycle time calculation,

	ProcessWeb	System Dynamics	Discrete Event
Role	1	1/2	1/2
Entity Attributes	1	1/4	1/2
Activities	1	1	1
Resources	1	1	1
Stat. Analysis	0	1/2	1
Time Horizon	Short	Long	Long
Interaction	1	1/2	1/2
Change Management	1/2	1	1

Table 1: Information on the characteristics of the different approaches for the study of dynamic aspects of processes.

we use the facilities (provided by the tool) of stamping the flow of material at different parts on the chain. In our case, we stamp the starting flow (*new agreements*) and the flow at the end (*process complete*). We also have a stock (*cumulative rework*) to account for the total number of agreements corrected.

DISCUSSION

We have showed that the structure of the OSCA process captured with RADs is maintained in our simulation and enactment models. This helps to promote the communication and understanding among the different agents in an organisation. On the other hand, we choose to develop the simulation models in graphical oriented simulators, that allow us to easily develop models. Thus, we can concentrate more on the analysis, evaluation and improvement of the processes in the organisations.

However, some of the information required for a consistent simulation model is missing in the RADs and also in PE. These are the following: i) they do not provide service times for the activities and arrival times, ii) no statistical information of any kind is provided (for instance, probability distributions of agreements that need to be corrected).

In table 1, we present some of the aspects of a process that the different simulation approaches provide. The scale of "1" implies that the approach provides that feature, and "0" that it does not. Clearly there are some degrees in between. ProcessWeb has been built around the concept of roles, however, SD and DE have not, but roles can be introduced as part of the simulation modelling. In the case of SD, it can be introduced through a sector frame diagram (large boxes in Figure 4). In DE, we can introduce labours (see man icon in Figure 3) associated to the activities of a role, therefore we create the role boundaries.

We consider the features on entity attributes to be the aspect where the approaches can make the difference in providing information. ProcessWeb can represent entities in terms of all the informational content, but in principle SD and DE do not provide it. Nevertheless, for purposes of entity manipulation, conditions under which an entity modifies its attributes, we can observe that DE provides more flexibility to do this sort of representations. Also SD provides some elements to manipulate or monitor the attributes.

In relation to activity representation and the resources to carry out the process, we observe that the three approaches provide them. In DE we use servers (machines in Wit) to represent activities, for SD we use conveyors, and for PE in the actions section within the role construct in PML. The aspect of resources is concerned with the people (agents) that perform each role. In DE, they can be introduced with labour components, that have to be available to carry out an activity. In SD, we can use support infrastructures to control the flows in the work-flow chain. For the PE, it is needed of a user playing a role to interact with the Web page to change the process state and consequently the page content. The statistical analysis feature is intended as a means of providing information on dynamic aspects (average time and size of queues, attribute behaviour, etc.) PE does not provide any because it focuses on individual agreements, and DE and SD have some elements for gathering this sort of information, as they focus on averages. Due to the nature of processes in organisations, we consider that DE is the most robust in providing statistical information. The time horizon refers to the number of times that a process can be performed with the purpose of a study. This is an important feature with respect to the use of simulation in process redesign or improvement. PE allows short time evaluation of a process whereas the DE and SD provide long time analysis.

An important aspect in a process is the interaction between roles. By construction PE in ProcessWeb has this feature, however in DE and SD it is not imposed but can be introduced in modelling terms. It is important to be aware of this aspect, because it is in these points or states, where the process timing can be a crucial factor for a process improvement.

Finally, the aspect of change, PE can be modified to cope with new situations, however, it does not have any element that could help us to evaluate how good the change is. For this reason simulation is a good complement and is being widely used in business process studies (Pegden 1997).

CONCLUSIONS

With this work, we intended to show that the techniques grouped together provide more information of the different aspects of a process and in general complement its overall description. It is important to construct from previous modelling, to avoid duplication of efforts and above all, to have a common ground of discussion and communication (RADs representation) for the main aspects of a process: functional, behavioural, organisational and informational (Curtis et al. 1992). This approach provides a much better analysis and dialogue on experiments and, results from a simulation study among the different agents in the organisation. The OSCA simulation model maintains the same structure that its static representation (RADs). We consider that this will help in the decision making and understanding of complex business processes. Simulation and enactment are tools that complement each other.

We also introduce, in the context of a simple process with two roles, some of the complementary information required in the mapping process from RADs to DE (entities, servers, queues, events), SD (flows, rates, levels, converters) and PE (roles, actions, entities, giveports, takeports). Even though, the model is very simple, it has most of the required characteristics (decisions, interactions, iterations) to highlight the features for mapping from one kind of representation to another. We would be able to take this ideas further in more complex models (Stock delay problems, processes with more roles and interactions, etc.)

Thanks to Mark R. Greenwood for his useful comments and discussions.

REFERENCES

Curtis, B.; Kellner, M. I. and Over, J., 1992, "Process Modeling", *Communications of the ACM*, 35, no. 9, 75-90.

Greenwood R. M. and Warboys B., 1996, "ProcessWeb-Process Support for the World Wide Web", In *Software Technology* Proceeding of the 5th European Workshop (EWSPT'96), C. Montangero (ed), vol, 1149 of LNCS, Springer-Verlag, Navicy, France, October.

Ould M. A., 1995, *Business Process: modelling and analysis for re-engineering and improvement*, John Wiley and Sons, Chichester, England.

Pegden, C. D., 1997, "Simulation: A Look to the Future", *11th European Simulation Multiconference (ESM'97)*, K17-K21. Richmond B., Peterson S. and Charyk C., 1994, "Introduction to System Thinking" and "Ithink guide", High Performance System Inc, Hanover NH, USA.

Swami A., 1995, "Building the Business Using Process Simulation", In *Proceedings of 1995 Winter Conference*, ed. C. Alexopouilos, K. Kang, W. R. Lilegdon, and D. Goldsman, 1081-1086.

Thompson B. W., 1995, "Introduction to WITNESS and Linking to Process Mapping Tools", In *Proceedings of 1995 Winter Conference*, ed. C. Alexopouilos, K. Kang, W. R. Lilegdon, and D. Goldsman, 462-466.

Walker A., Kawalek P. and Greenwood R. M., 1996, Lecture notes of course *Process Modelling*, Dept. Computer Science, University of Manchester, UK.