Integrated System and Process Analysis

The Right Technique for the Right Problem

Henk Jonkers, Wil Janssen, and Henry Franken

During the design of distributed systems and (business) processes, model-based analysis of both functional and quantitative properties plays an important role. Depending on the required level of detail, the analysis is either performed using analytical methods or simulation. At the Telematica Instituut, a modelling framework has been developed providing a uniform interface for different types of analysis. This framework can be used to support the design of a wide variety of systems of interest to the institute, ranging from network infrastructure to business processes and cross-organisational co-operation.

Model-based analysis and simulation play an increasingly important role in the design of a great variety of systems, as models often serve as a blueprint for the actual implementation. Case-tools and workflow systems, for example, transform a model to an operational system. The properties of the model thus become properties of the real sytem. Looking at the research areas of interest to the Telematica Instituut, these systems range from computer hardware and communication networks to complex business processes (or even cross-organisational processes) supported by information and communication technology. Functional analysis is used to validate the correct behaviour of a system, while quantitative analysis provides an instrument to, e.g., find bottlenecks, compare alternative designs or fine-tune the performance of a system. It is important to use analysis throughout the design process. All too often, a completed design is analysed as an “afterthought”, to check if the functional and quantitative (quality of service) requirements have been met. The risk of this approach is that an expensive redesign of the whole system might be needed if it turns out that the system behaves incorrectly or the performance is insufficient. In this paper we present some examples of how modelling, analysis and simulation techniques are used to support projects that are carried out in the Telematica Instituut. We first give an overview of possible approaches to system analysis.

Three ways to derive system properties

Basically, three classes of methods can be applied to derive estimates of system properties and measures:

- measurement,
- simulation, and
- analytical methods.

Measurements (or, more appropriate for functional properties, *observations*) are only possible if the system of interest is already operational. Therefore, they will be of limited use during a (re)design, because they cannot be used to obtain *predictions* of the system behaviour. Measurements provide *on-line* analysis, as opposed to the other two methods that are used for *off-line* analysis.

Simulation could be described as the direct execution of a model, expressed in either a special-purpose simulation language or a general-purpose programming language. It is a powerful instrument, as it can be used to study any system aspect to any level of detail. However, simulation is time-consuming, which makes it less suitable to quickly compare a large number of design alternatives. Also, because nearly all models contain sources of non-determinism, simulation typically *provides* probabilistic results rather than unique, reproducible predictions of the system properties.

Analytical solution techniques derive quantitative measures (either exact or an approximation) in a systematic, mathematical way. They can be subdivided in *symbolic* techniques and *numeric* techniques. Analytical solutions are commonly used for quantitative analysis. Also for functional analysis such techniques are available: state-space exploration techniques such as model checking [3] ensure that all possible scenarios in the system are checked. A drawback of analytical techniques compared to simulation is that the class of models that are analytically tractable is limited. For model checking the size of the model is the most important bottleneck.

Although this rough classification is useful to place the different techniques, it is not always possible to draw a sharp line between simulation and analytical techniques. Some numerical analysis methods closely resemble a simulation approach, and certain kinds of simulation, e.g. Monte Carlo methods, are only a small step removed from analytical methods. Moreover, simulation and analytical methods can be used in combination: submodels that are analytically untractable are solved by means of simulation, after which the overall model is solved analytically, or vice versa. In functional techniques such as model checking, counter-examples of a checked property can often be visualised by means of a stepwise simulation. Thus, instead of two completely disjoint approaches, we obtain a more or less continuous range from purely analytical methods to pure simulation.

It is important to realise that there is no "best" approach to functional or quantitative analysis: simulation is not inherently better or worse than analytical techniques, and the same applies for different analytical techniques.

The Telematica Instituut [http://www.telin.nl] is a market-driven research institute administered and financed by leading enterprises and supported by the Dutch government as a top technological institute. The institute's main goal is to quickly translate fundamental research into commercial applications in the field of telematics. In collaboration with many (inter)national knowledge centres we work on strategic top research for businesses. The projects are typically multi-client and multi-disciplinary. Our expertise areas are:

- networked electronic collaboration (e.g. computer supported cooperative work and groupware applications);
- networked electronic commerce (e.g. business network design and supporting transaction systems);
- content management (information retrieval, intelligent agents, knowledge management and teletraining);
- middleware (e.g. architecture and technology building blocks for next generation internet).

The Telematica Instituut is managed from the central organisation in Enschede.

The main difference between the approaches is their position on the natural trade-off between the accuracy of the analysis results and the computational complexity. Symbolic analytical results provide very efficient first-order estimates of performance measures, and are typically used in the early design stages to support the choice between major design alternatives. Detailed simulation provides accurate results, but it is relatively time-consuming. It is typically used to fine-tune the performance of a final design.

Our approach to system analysis

Especially for simulation, the number of existing tools, both general-purpose and specific for a certain application area (e.g. manufacturing systems or business processes), is enormous. But also for analytical techniques for functional or quantitative analysis the available tools are constantly improving and increasing in number. Therefore, it is useless to build "yet another" simulation tool. Our interest is primarily how to deploy existing tools, possibly customised to our specific needs, to optimally support the design of systems and processes. However, we do not want to burden designers with the particularities of all these analysis techniques, modelling formalisms and tools.

We solved this problem in the context of business process (re)design by means of a common modelling and design language that, in addition to representing processes in an insightful way, serves as a front-end to multiple analysis methods. The same approach can be used in other application areas. Figure 1 summarises our approach. A "system" (e.g. business process or distributed application), consisting of a number of resources and a number of corresponding processes, is modelled in an analysis-independent modelling language. The model, consisting of an actor model and a behaviour model, is mapped to an analysis-specific modelling formalism, either quantitative or functional (or both, e.g. timed Petri nets). Analysing this model, either by means or simulation or an analytical method, yields estimates of process-oriented or system oriented measures, or of functional system properties. These results can be used as feedback to the original model and ultimately the modelled process or design. With this approach, simulation and analysis becomes more accessible to system designers with limited knowledge of these techniques.

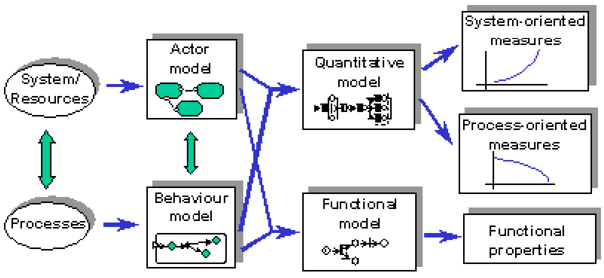


Figure 1. Overview of the analysis approach

We illustrate the approach by an example in the field of business process redesign (BPR).

Business process analysis

In the Telematica Instituut, simulation and analysis techniques have been applied most extensively in the Testbed project [http://www.telin.nl/testbed], which develops a language, methods and a software tool (called Testbed Studio) to support business process modelling and (re)design in the financial sector [2]. Project results have been applied in several real-life business cases. The project is a collaboration of the pension fund ABP, the Dutch Tax Department, ING Group, IBM and the Telematica Instituut, and is financially supported by the Dutch Ministry of Economic Affairs.

One of the principle ideas of the Testbed approach is that a single graphical modelling language is used as a common front-end for different types of analysis. This language has been designed in such a way that clear process models or specifications can be constructed, which are easy to understand and to communicate to all parties involved in the process (re)design. At the same time, a sound mathematical basis of the used concepts guarantees the possibility of an unambiguous mapping to the different analysis formalisms. In this way, we diminish the traditional gap between design languages and (functional or quantitative) analysis formalisms.

Figure 2 shows a simple example model to illustrate our modelling language. It models the first steps of the process of a car damage claim handling. The model consists of two aspect models, a *behaviour* model representing the activities in the process and their relations, and an *actor* model representing the parties (in this case people) involved in the process. Activities are denoted by (stretched) circles (with a shadow for replicated actions), and arrows between the actions define the order in which these activities must take place. Parallelism is modelled with *and*-splits (black diamonds) and *and*-joins (black boxes), while choice is modelled with *or*-splits (open diamonds) and *or*-joins (open boxes). The trigger (*report damage*) starts the process and can be used to, e.g., specify the arrival rate of claims. Resources are denoted by octagons, and interaction points (connected ovals) show which actors can have direct interaction. A process model and an actor model are linked by means of an *actor attribute* of an action, which specifies the resource that performs the action.

Simple step-wise simulation and analytical quantitative methods are incorporated in the tool, and more complex types of analysis, e.g. extensive quantitative simulation, are realised by means of interfaces to other special-purpose tools that are available.

Functional analysis is used to validate process properties: e.g., referring to the example process, can we guarantee that no report is forwarded without having consulted at least one witness? In this simple example it is obvious that this is not the case, but in more complex processes, with many alternatives and parallel paths, it is often non-trivial to answer such questions. Testbed Studio includes a built-in stepwise simulator, which can be used to "step" through a process model, thus obtaining insight in the possible sequences of actions that can occur. However, for a definite answer we need a technique like *model checking*, which searches the whole state space for possible counter-examples of the required property. In accordance with the "lightweight" requirement, Testbed Studio provides an interface to the model checker SPIN [3]. This is done in such a way that the use of SPIN is hidden from the end-user, which means, among others, that a user-friendly mechanism has been developed to specify the process properties to be checked.

With *quantitative analysis*, we want to obtain insight in process-oriented measures such as completion times and waiting times and resource-oriented measures such as resource utilisations. These are typically used to identify process bottlenecks or to compare alternative processes to optimise a design. Comparable to functional analysis, a number of relatively simple analytical techniques, used to derive fast first-order estimates of the quantitative measures, are incorporated in Testbed Studio. For the process-oriented measures, critical path analysis is available for a very efficient estimate of the *mean* completion time of a process, but also to find out which actions make up the critical path, and are likely candidates for improvement. The *probability distribution* of the completion time can be computed with another analytical technique, based on stochastic graph reduction. This technique typically results in a graph as shown in Figure 3, and in addition to a more accurate estimate of the mean completion time it allows us to answer questions such as "Which percentage of the received damage claims is completed within the norm of eight days?" Finally, queueing analysis is available to derive mainly resource-oriented measures, e.g. resource utilisations and the mean number of waiting customers. Critical path and queueing analysis can be used in a combined way. This provides a powerful and efficient instrument for a joint study of process-oriented and resource-oriented measures and their interrelation.

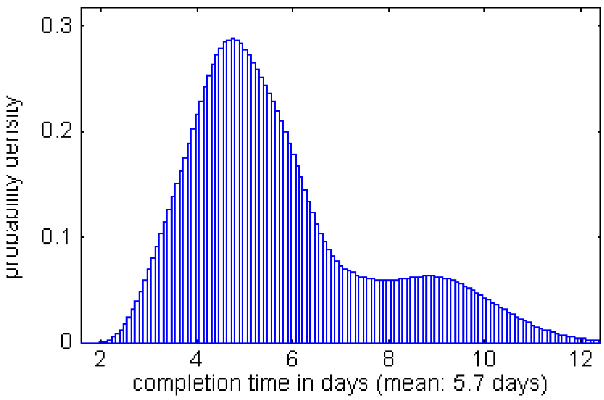
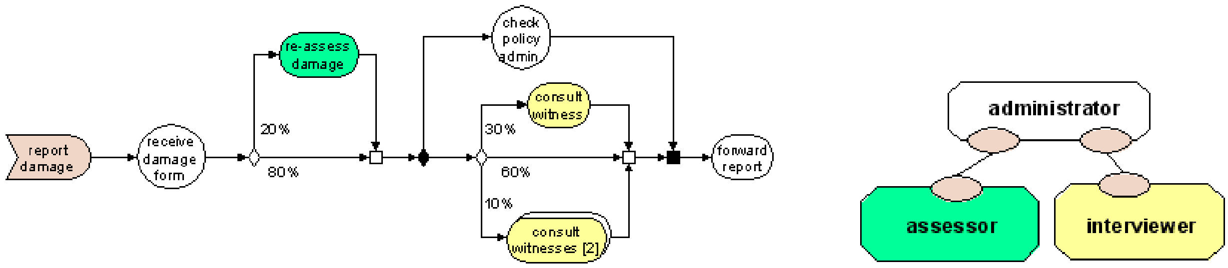


Figure 3. Example of analysis results

Although analytical methods are very useful to obtain fast first-order estimates of the quantitative measures, their applicability is inherently restricted. They are relatively inaccurate, which means that they may not suffice for fine-tuning a process. Moreover, a process might be too complex to be captured in an analytically tractable model. Typical examples of the latter situation are complex resource allocation strategies or complex synchronisation patterns between activities. Also dynamic decisions, e.g. a number of available resources that varies over time and depending on the workload, fall into this category. In those cases, we have to resort to quantitative simulation. In the light of the huge number of specialised tools that bexist for this purpose, the obvious choice is once again to provide an interface to one of these tools rather than to develop one of our own. In collaboration with the School of Systems Engineering, Policy Analysis and Management of Delft University of Technol-



behaviour model   actor model

Figure 2. Simple business process model

ogy, some candidates for this coupling are assessed, and a prototype will soon be available.

Other applications

Although developed in a business process modelling context, many of the ideas from the Testbed project can be more widely applied. The results of the project have now reached a stage that they can be deployed in other projects and other application areas of interest to the Telematica Instituut. For example, we use similar models to analyse cross-organisational transactions, e.g. e-commerce transactions, in the NETS project [1].

The most obvious application that comes to mind is the design of distributed systems or applications, where it is often non-trivial to determine whether the complex combination of hardware and software components and network infrastructure provides the required functionality and performance. A similar approach, albeit not yet based on the Testbed language, was applied to advise Rijkswaterstaat on the network capacity needed to meet the performance requirements of messages sent from roadside systems to their traffic centres.

In new telematics applications to be developed for next generation internet, among others within the GigaPort programme [http://www.gigaport.nl], this will become increasingly important. The basic modelling concepts that were used as a starting point for the Testbed language originate from distributed system design [5]. Also, most of the described functional and quantitative analysis techniques find their origin in the validation or performance analysis of computer and telecommunication systems. Therefore, it is not surprising that the Testbed results translate to this field in a natural way. In [4] we illustrated this with a simple distributed image database example.

Conclusion

In this paper we illustrated how modelling and analysis, either based on simulation or analytical techniques, are used within the Telematica Instituut to support the design of systems or (business) processes. With our approach, simulation and analysis becomes more accessible to system designers with limited knowledge of these techniques. Thus the risk of using analysis as an afterthought only becomes smaller: an important obstacle for integral analysis as part of the design process is elminated.

References

[1] F. Biemans, W. Janssen, P. Oude Luttighuis, H. Schaffers and P. van der Stappen, “Business-driven design of transaction services”, 5^{th} *International Conference on Concurrent Enterprising*, The Hague, March 1999.

[2] H.M. Franken and W. Janssen, “Get a grip on changing business processes: Results from the Testbed project”, *Knowledge & Process Management* (J. Wiley), winter 1998.

[3] Holzmann, G.J., “The model checker SPIN”, *IEEE Tr. on Software Engineering* 23: 5, May 1997, pp. 279–295.

[4] H. Jonkers, W. Janssen, A. Verschut and E. Wierstra, “A unified framework for design and performance analysis of distributed systems”, in

Proc. 3rd Annual IEEE Int. Computer Performance and Dependability Symposium, Durham, NC, Sept. 1998, pp. 109–118.

[5] D.A.C. Quartel, L. Ferreira Pires, M.J. van Sinderen, H.M. Franken and C.A. Vissers, “On the role of basic design concepts in behaviour structuring”, *Computer Networks and ISDN Systems* 29: 4, March 1997, pp. 413–436.

Author biographies

Pasfoto H. J.

Dr. ir. Henk Jonkers received his M.Sc. in Computer Science from the university of Twente, and did Ph.D. research on performance analysis of parallel and distributed computers at Delft University of Technology. Since 1995 he has been a member of the scientific staff at the Telematica Institute. His main research interests are business modelling, analysis and design, and quantitative methods.

Dr. ir. Wil Jansen studied Computing Science at Eindhoven University of Technology. After his Ph.D. on design of distributed systems at the University of Twente, he was a postdoc on formal methods at the University of Oldenburg. Since 1996 he has been a member of the scientific staff at the Telematica Institute, and was involved in modelling, analysis, and design of business processes. His current interest is support for the design of telematics services for cross-company co-operation.

Pasfoto W. J.

Pasfoto H. F.

Dr. ir. Henry Franken received an M.Sc and Ph.D (with honours) in Electrical Engineering from the University of Twente. He is a member of the scientific staff at the Telematica Institute and manager of the Testbed project. His research and consulting activities focus on applying system engineering principles to telematics amd business process (re-)design.