Modeling and Evaluation of Time Aspects in Business Processes

Juliane Dehnert[1], Jörn Freiheit[2], and Armin Zimmermann[2]

[1] Kommunikations- und Softwaretechnik
Technische Universität Berlin, Sekr. E-N 7
Einsteinufer 17, 10587 Berlin, Germany
[2] Prozeßdatenverarbeitung und Robotik
Technische Universität Berlin, Sekr. E-N 10
Einsteinufer 17, 10587 Berlin, Germany
{dehnert|freiheit|azi}@cs.tu-berlin.de

Abstract. There is a need for modeling and performance evaluation techniques and tools for a fast and reliable design of workflow systems. This paper introduces a modeling methodology based on colored stochastic Petri nets. It allows the integration of control flow, organizational, information related and timing aspects in one modeling framework. The processing delays include stochastic distributions in addition to deterministic times. Several workflows and the effects of constrained shared resources needed for different tasks can easily be described and analyzed together. Control flow and organizational aspects are modeled separately in resource and workflow models. These models are automatically compiled into one model, which can then be used for qualitative analysis or performance evaluation. The proposed modeling and evaluation method is supported by the software tool TimeNET. An application example shows its use.

Keywords: Performance Evaluation, Stochastic Petri Nets, Simulation, Insurance Case Study[1]

1 Introduction

Because of the increasing demand for business process (re-)engineering, business processes and their design have gained much interest recently. Many companies around the world try to reorganize their business processes in order to gain competition advantages. This development increases the demand for tools and techniques to support the identification, planning, and evaluation of business processes. It is clear that this can only be done using a more or less formal and abstract description of the process, namely a *business process model*, that serves as a basis for further analysis. Such a model of a business process reflects the

[1] Journal of the Operational Research Society **53**, pp. 1038-1047, 2002.

control flow between tasks carried out by the organization as well as the correct document processing therein.

The methodology presented in this paper is based on Petri nets, a graphical representation of systems including synchronizations and concurrency. Petri nets have been applied to numerous areas successfully, among them business processes. The suitability of Petri nets for this field of application has been examined and discussed extensively in the literature.[1,2]

Some important requirements for this application domain are discussed in the following. During the identification of business process steps as well as during later reorganizations the business process model is the common basis of communication. Thus it has to be easy to understand for non-experts as well. Petri nets allow a simple graphical description of tasks, states and interdependencies. Their formal foundation avoids misunderstandings.

For problems of real-life size a compact representation of complex structures as well as hierarchical decomposition are of particular importance. The type of colored Petri nets used in this paper supports these requirements.

From a general viewpoint the following different aspects need to be incorporated in a business process model. *Control flow* aspects (Which tasks? In what order?) are complemented by *organizational aspects* (By whom? By what tool?). Information related aspects (*data flow*: Which items/documents are processed?) are a further essential.

Most existing approaches restrict their view to only one of the mentioned aspects. The approaches proposed in[3,4] concentrate on the modeling of control flow aspects only. Other aspects are neglected. Therefore these methods are not able to capture the influence of shared resources and the interdependencies between different concurrent workflows.

In the remainder of the paper we will introduce a methodology for the modeling and performance evaluation of workflow processes, which is not limited to one aspect. It describes control and data flow as well as organizational aspects. The model is based on a concept of independent models for the control flow and organizational aspects, both using a dedicated class of colored stochastic Petri nets. The control flow aspects (possibly of different workflows) are specified in the *workflow models* and the organizational aspects in the *resource models*. A workflow model describes the possible executions using the resources described in the resource models. Data flow is integrated in the model through different tokens which correspond to processed documents.

The efficiency or performance of a business process is of special importance during a workflow analysis. Example projects could try to reduce execution times in order to improve competitiveness and customer contentedness. Different possible organization structures or staffing options can be compared using an evaluation of the business process performance. Performance evaluation techniques can be applied to a business model. This requires the model to be executable, meaning that its behavior is clearly defined.

In contrast to the *critical path technique* or *event process chains* Petri nets describe the dynamic behavior of systems, because they are directly executable.

In this case simulation and analysis algorithms can be applied. Analysis or simulation support to predict and optimize the performance of the modeled organization and its business processes. Example objectives could be increasing the degree of service to the customer, or find the optimal number of staff and resources needed.

Another prerequisite for evaluation is a notion of time in the model. It must be possible to specify task durations as well as deadlines. Although this seems to be a crucial point for business process engineering, it has not gained much interest yet. There are only few approaches[5–9] considering time, offering analysis or simulation as a means to evaluate the behavior of the workflow model. Often the notion of time is limited to the control flow view and to deterministic delays and time constraints.

Stochastically distributed durations are mostly neglected. However, in the field of workflow performance modeling a mixture of both stochastic and fixed (deterministic) durations are necessary. Fixed times are needed to model durations of automated processes, deadlines, and routine activities with previously known delay. Stochastic durations are necessary to model human activities, stochastic events like employee absence due to sickness or holidays, and other events with unknown duration. Consequently, deterministic and stochastic durations should be integrated into the model.

In the approach presented here, evaluation of the performance is facilitated by associating stochastic, deterministic, or zero firing delays with transitions. Basic quantitative measures like the throughput, utilization, queue length, processing time, and others can be computed either by direct numerical analysis or discrete event simulation. To the best of the authors knowledge, the approach presented here is the first to incorporate both stochastic timing aspects (allowing performance evaluation), and the full specification of all workflow aspects. In that, it aims at bridging the gap left by prior approaches.

To support the proposed methodology, a software tool is necessary. Throughout this paper the tool TimeNET is used,[10] in which the described modeling and evaluation functions are implemented. The proposed methodology is demonstrated using an industrial case study. The considered business process describes the handling of damages and losses in a transport company.

The remainder of this paper is organized as follows: Petri nets are briefly covered first. In Section 3 we then introduce the application example. Section 4 explains the modeling technique, while performance measures and their computation are covered in Section 5. The subsequent Section 6 contains information about software tool support for the proposed method. Finally, concluding remarks are given.

2 Petri Nets

Petri nets[11,12] are based on the ideas of C.A. Petri. They are used to model and analyze complex concurrent and distributed systems. The attraction of Petri nets is based on the ease of conceptual modeling using a simple graphical notation and the mathematical background allowing to formally capture many of the basic notions and issues of concurrent systems.

The most basic are the notions of *local states* and *local actions* and their mutual relationship. Local states are graphically represented by circles and called *places* while local actions are called *transitions* and depicted by rectangles. In general, a local action is influenced by a subset of local states and influences a subset of local states. These influences are graphically represented by *arcs*.

To model and analyze the dynamic behavior of Petri net models, so called *tokens* are used to describe the set of 'current' local states. They are depicted by black dots inside the places. A global state is the union of all local states.

Figure 1 shows a small Petri net example modeling a process within a travel agency.

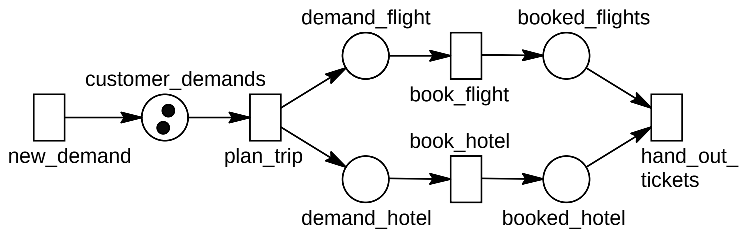

Fig. 1. Petri net model example: "Booking a Trip"

There are two tokens in place `customer_demands`, which is thus called *marked*. Marked places represent current local states of the modeled system. A transitions may change a local state. It is enabled if all input places are marked. An enabled transition may fire. In the example transition `plan_trip` and `new_demand` are enabled. The firing of a transition consumes one token per input place and produces one token per output place. Firing of transition `plan_trip` consumes one token from place `customer_demands` and produces one token on `demand_flight` and `demand_hotel` each. Thus, a global state is reached where the place `customer_demands` contains one and `demand_flight` and `demand_hotel` contain one each. In the new state all transitions except `hand_out_tickets` are enabled.

There are many ways to enhance the basic concepts of Petri Nets. An important extension is the introduction of *colored*[13] tokens. This concept allows to distinguish between token. They are not 'black' anymore but consist of several describing items (at least one identifier). Figure 2 depicts a colored Petri net. The model describes the same process but distinguishes between different customer demands, which are treated differently.

The Petri nets class presented in this paper is a dedicated class of colored Petri nets.

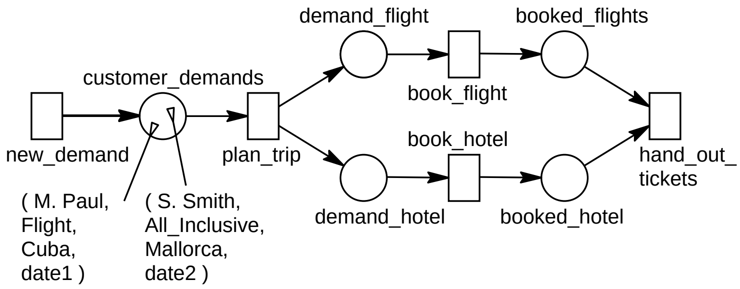

Fig. 2. Resource model of a trip booking system

This class supports the modeler to distinguish between resource and workflow models. The resource model depicts the structure of the system with all possible ways of handling a case and the available resources. In contrast to that the workflow model describes the handling of one instance of a case. It can be understood as the path of a specific token through the resource model.

Figure 2 shows the resource model of the trip booking example while figure 3 presents an associated workflow model. It describes the specific handling of a customer demand requiring a flight only.

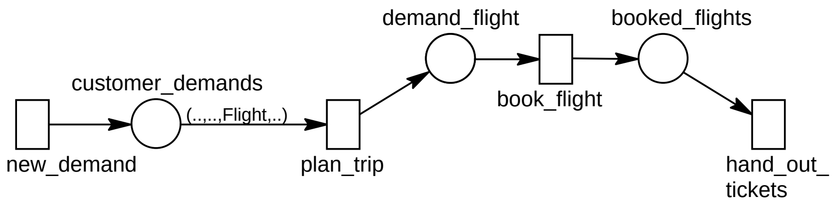

Fig. 3. Workflow model for handling a customer demand requiring a flight

A further important extension concerns the integration of time. This allows to evaluate the performance of the modeled systems. In *stochastic* Petri nets,[14–16] which are used in this paper, each transition has an associated distribution function describing their firing delay. With respect to the example an exponential firing delay is used for transition `new_demand` while a deterministic delay is assigned to the transition `book_flight`. This models a stochastic inter arrival time of new demands and a fixed duration of the flight booking activity.

3 An Application Example

This section presents an industrial case study. A damage event handling system is modeled.

In a damage event handling system there are many persons and institutions involved. Planning and managing the corresponding business process and its resources is a very complex task. Many concurrent activities have to be synchronized.

Figure 4 gives a rough overview of the involved offices and organizations as well as possible information flows of the transport company. It should be noted that this is only intended as an informal way of describing the organization.

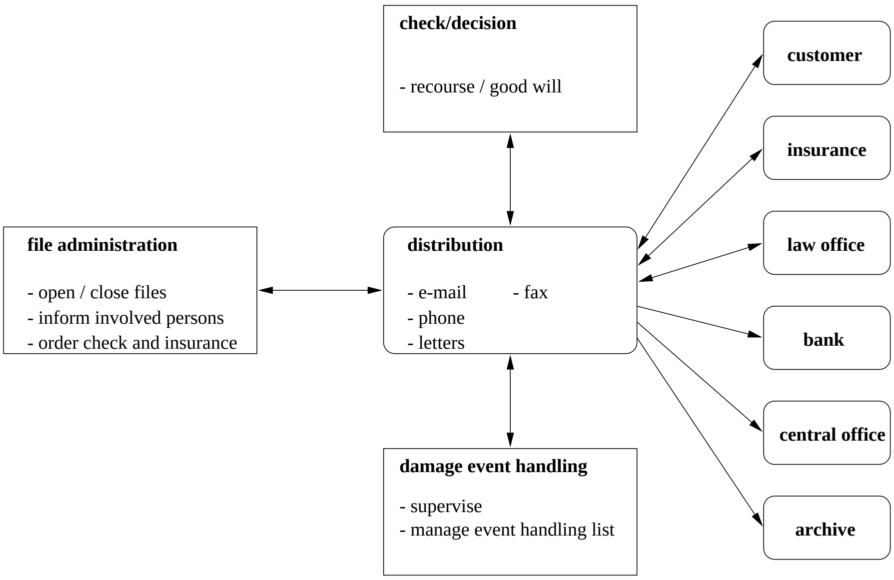

Fig. 4. Coarse structure of the damage event handling system example

In the example three offices are considered in detail, namely file administration, check/decision, and the main damage event handling office. In Figure 4 they are depicted by rectangles. The damage event handling office supervises the business process that is started for each incoming message of a damage event. An event list is managed for each open case by this office to keep track of its progress. The file administration office processes the associated files, informs the involved persons, orders checks, and passes the necessary information to the involved insurance companies. The check/decision office checks the case and decides whether the damage event is treated as a recourse or a fair dealing.

All necessary communication between the involved offices and the exchange of documents between them is depicted as distribution. The distributor can be an e-mail server, a phone or fax system, a mail office or a combination of them. In the following we abstract from the actual technical way of communication for simplicity. The information distribution also serves as the interface to the environment of the company, drawn as small boxes with round corners. Arcs in the figure describe possibilities of communication.

In addition to the organizational structure the workflow processes of the example are described in the following. Figure 5 shows a part of the example workflow.

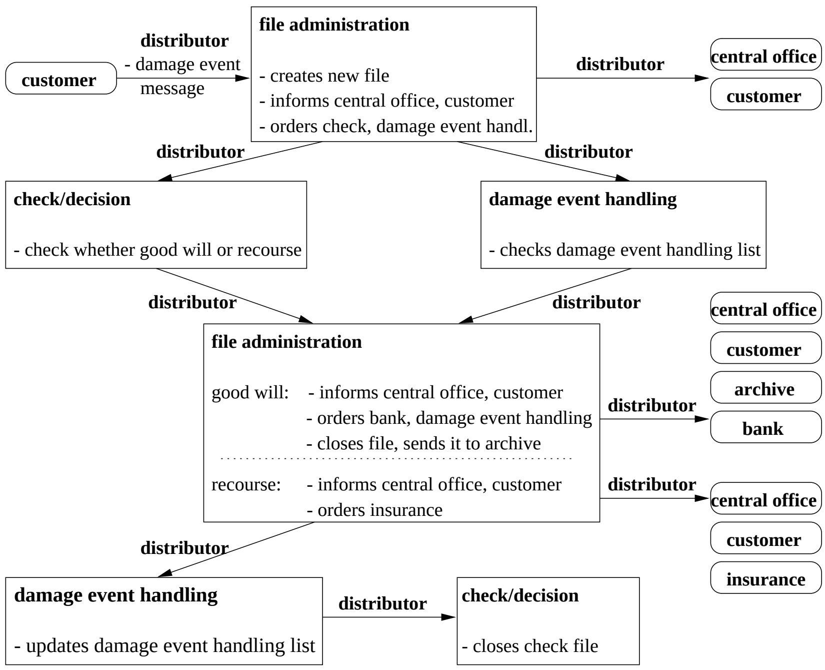

Fig. 5. Part of workflow in the system example

The workflow is initiated when a customer message arrives, informing about a new damage event. The file administration office creates a new file, informs the central office and the customer via distributor, and orders an analysis of the case by the check/decision office. The damage event handling office is informed in parallel, checks the damage event handling list and sends it to the file administration office. The check/decision office decides whether the damage event is treated as a recourse or a good will. If it is a recourse, the insurance has to pay for the damage, and is therefore informed. In the case of a fair dealing the company orders its bank to pay the damage itself.

In Figure 5 only a selected part of the whole workflow is depicted, which is later also shown as a PN model. The workflow for the recourse case is omitted. If the check/decision office decides that the insurance has to pay, but the insurance decides otherwise, an additional check of the case is done. If the decision is maintained, the case will be transferred to the law office of the company. Another detail not shown in the figure is that if the insurance did not pay after some predefined delay, the file administration will send a reminder.

At the end of the workflow the file administration office closes the file and sends it to the archive. The damage event handling office updates the damage event handling list and finally informs the check/decision office about the closing of the case.

4 Workflow Modeling with Colored Petri Nets

In this paper a special class of colored Petri nets is used for the modeling of the workflow management system example described above. Two kinds of models are distinguished. The *resource model* describes the abilities and workflow independent properties of the workflow management system resources, such as communication connections, staff, technical resources etc. The *workflow models* specify the actual business processes that take place for every case. They describe assigned activities and the controlflow between them. Each workflow model can be thought of as a path through the resource model. Later on, the different model parts are automatically merged resulting in a complete model, which then includes both the resource constraints of the system and the synchronization of the workflow activities. The models are hierarchically structured, which is necessary to handle complex systems.

As it is usual for Petri net models, transitions model activities of the system, while places are passive elements and contain tokens that model moving and/or state changing entities. Transitions with thick bars are called substitution transitions, acting as place holders for submodels describing their behaviour in more detail on a lower level of hierarchy. Places shown as dotted circles connect the submodel with its environment. Transitions depicted as a bar fire immediately without delay. Transitions drawn as empty rectangles have an exponentially distributed firing time, while transitions with deterministic delay are depicted as filled rectangles.

Two colour types are predefined in the model class: *Object tokens* model files, orders, letters etc. inside the workflow management system, and consist of a name and the current state. *Elementary tokens* cannot be distinguished, and are thus equivalent to tokens from uncolored Petri nets. They are used to model states of the resources, for instance whether an employee is busy or not. Places can contain only tokens of one type. Therefore it is possible to graphically separate object places and arcs (drawn thick) as opposed to thinly drawn elements that correspond to elementary tokens. Textual descriptions needed in colored Petri nets for the definition of variables and colour types can thus be omitted, and the specification of the types of places and arcs are implicitly given.

4.1 The resource model

The resource model contains all used resources, like employees, computers, data bases, etc. and abstract possible actions of the resources, even if they are not used for the processing. Figure 6 shows the highest level of the hierarchical colored model.

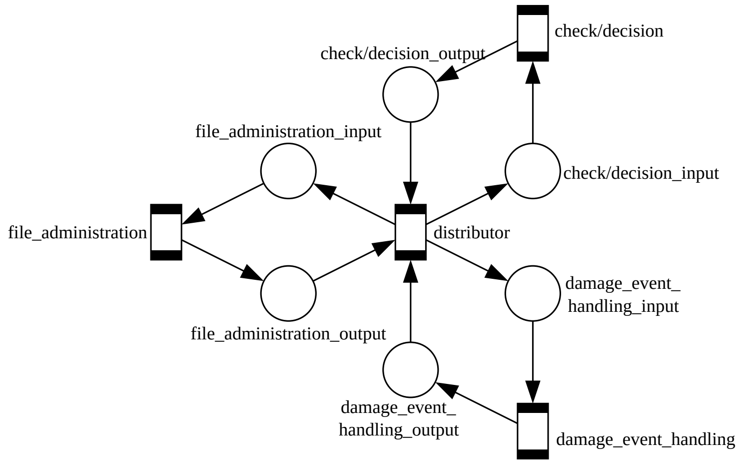

Fig. 6. Resource model of the damage event handling system example

Its structure follows the layout of the modeled organization, which makes it easier to understand. Places model possible locations of objects, like files, orders, letters etc. All offices exchange documents via `distributor`. The `input` and `output` places model in- and outgoing matters of the corresponding offices. In principle, there are two different operations that can be performed: transport and processing of objects. The former corresponds to moving a token to another place, while the latter is modeled by a change in the color of the token that corresponds to the object.

Figure 7 shows the resource submodel of the file administration office in more detail. The model is a refinement of transition `file_administration` in Figure 6. It has two parts. The transitions and places connected by thin arcs describe the behaviour and states of the employees working in this office. The place `employee_working` models the working employees. In this example there are two employees associated to this office, modeled as two token in the corresponding place. The `work_off` transition models the case that an employee goes to holidays or becomes ill. This interrupts the current work of the employee (transition `interrupt_process` fires), or only decreases the number of available idle employees (transition `end_of_work` fires). If the employee comes back to work (`work_on`), he is available again (`employee_available`).

The places and transitions connected by thick arcs describe the behaviour of objects like documents or memos. The dotted places connect the submodel with its environment. They correspond to normal places in a net of a higher level

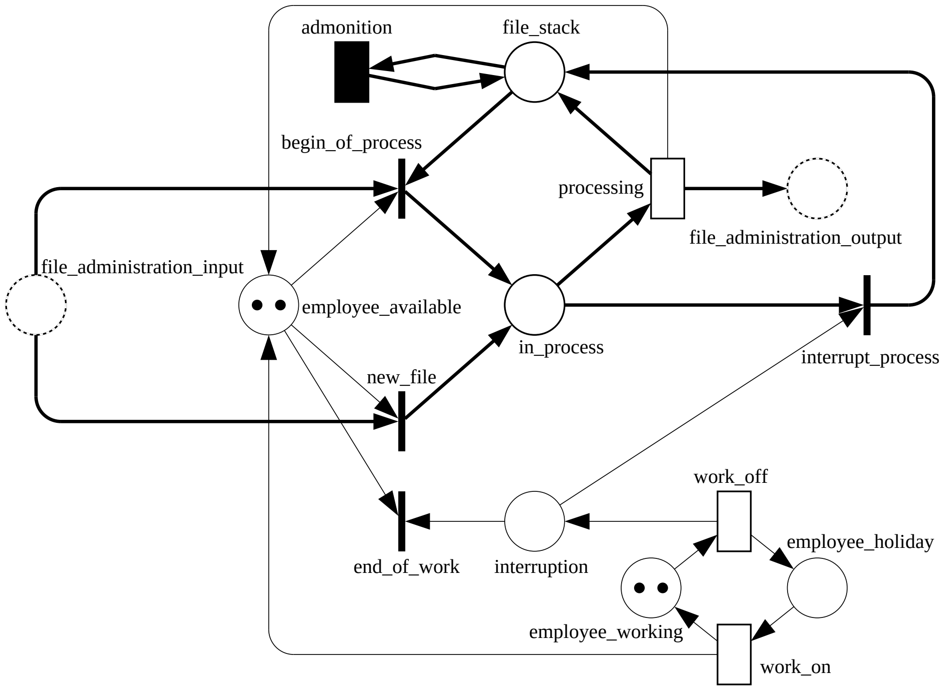

Fig. 7. Resource submodel of the file administration office

within the refinement hierarchy. Here the places `file_administration_input` and `file_ administration_output` correspond to the same places in Figure 6).

The place `file_stack` contains all files which are currently not being processed, while place `in_process` contains the active ones. The processing of an existing file starts with the firing of the immediate transition `begin_of_process`. For each new damage event message the transition `new_file` creates a new file. The transition `processing` describes the processing of a file. Due to the modeling of human actions in this case the transition has an exponentially distributed delay. The deterministic transition `admonition` fires after a fixed time and thus models deadlines for expected answers from other offices.

For the chosen application example the modeling of the different offices resulted in submodels that are very similar to the one described above. The whole resource model contains four submodels, one for each office and one for the `distributor`.

Transition delays specified in the resource model are standard values for the modeled actions. Depending on the actual workflow tasks the duration can of course differ from this value. Such specialized durations are associated to the transitions of the workflow models (see below).

4.2 The workflow models

In the resource model the structure of the system and possible ways of communication is summarized. The workflow model now describes the actual processes using the resources described in the resource model. Figure 8 shows the top level workflow model of the example.

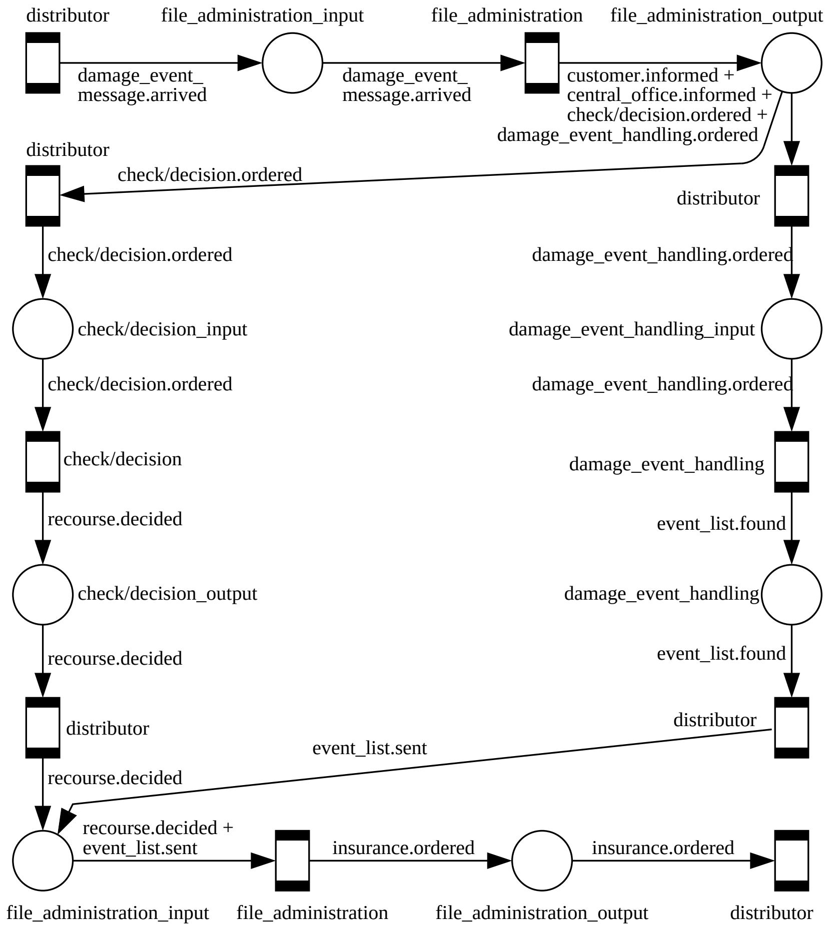

Fig. 8. Part of the top level workflow model of damage event handling

The workflow models describe paths through the resource model (see Figure 6). In the workflow model the arcs are inscribed by the tokens which flow through the arcs. After a damage event message has arrived via `distributor`, the `file administration` creates a new file and informs the customer and the central office. Additionally it sends the necessary information to the check/decision office and the damage event handling office. After the `check/decision` two cases are distinguished: Either the damage event is a recourse or a fair dealing. In Figure 8 only the recourse case is depicted. The check result `recourse.decided` is sent to the `file administration` office. The `damage event handling` office sends the damage event handling list to the `file administration` office. For simplicity, Figure 8 only shows the model until the point where the insurance is informed about the recourse case.

Figure 9 shows the workflow submodel of an arriving damage event message to the file administration office (see top of figure 8 transition `file_administration`). Workflow models are hierarchically refined in the same manner as the resource model. The relation between model elements of both parts is ensured by using identical names e.g. for a transition.

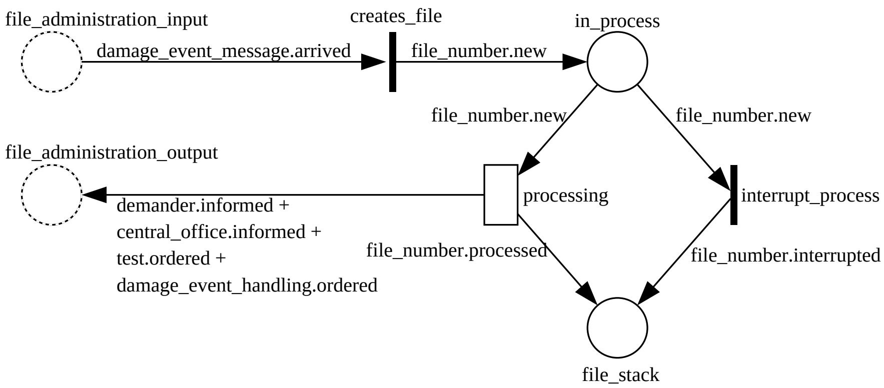

Fig. 9. Workflow submodel of arrived damage event message

The immediate transition `creates_file` takes the token `damage_event_message.arrived` from place `file_administration_input` and transfers the token `file_number.new` to the place `in_process`. The part of the token name before the dot describes the modeled object (e.g. `damage_event_message`). The second part of the token name (e.g. `arrived` or `new` describes the state of the token. The transition `processing` changes the state of the token `file_number.new` into `file_number.processed` while it creates four new tokens and puts them into the place `file_administration_output`. If the process is interrupted (e.g. because the employee is ill; see Figure 7) the state of the file will change from `new` to `interrupted`. Using the states of the tokens it is possible to model priorities, e.g. to first finish an interrupted case before starting a new one. In the example tokens with state `interrupted` have higher priority than `new` or `processed` ones.

5 Evaluation of the Workflow Model

The previous sections described how to construct models of the workflows and used resources. Subsequently, both model parts are merged automatically to create a complete model that can be analyzed. The information contained in the workflow models is added to the structure model during this process. Invisible for the modeller, the transitions are enriched with their *firing possibilities*.

5.1 Compilation of a Joint Model

In a colored Petri net, each transition may have several firing possibilities depending on the current model state (the marking). Each is characterized by different values of the arc expressions attached to the transition's input and output arcs. In contrast to other colored Petri net modeling techniques, all different firing possibilities can be obtained automatically from the workflow models. The set of token colours is given by the object tokens used in the inscriptions plus the predefined type of elementary tokens. Each of the workflow models describes the processing steps of one task. Thus, every occurrence of a transition in such a model specifies the use of a resource that exists in the structural model.

Therefore each occurrence of a transition in a workflow model describes one possible activity of the resource and results in one transition table entry of the compiled model. This table is just a collection of the firing possibilities of each transition of the resource model. The compiled model consists of the resource model, together with the transition tables. A more detailed description of this algorithm (originally introduced for manufacturing system models) can be found in.[17]

5.2 Qualitative Analysis

In order to find modeling errors or general design faults without a full analysis of the reachability graph, several techniques are applicable. First of all the *token game* can be executed to ensure that the model works as intended. Combined with an animated visualization of the behaviour this leads to a better understanding of the workflow system.

Structural analysis techniques allow the computation of model properties like invariants, conflicts, and other information about the Petri net model that can be interpreted in terms of the workflow system. It is e.g. possible to check whether the number of employees is constant inside the model by computing place invariants.

5.3 Performance Evaluation

The behaviour of the model is given by the initial marking and the subsequent transition firings, describing a stochastic process.[18] The type of process depends on the types of allowed firing delays and whether certain transitions are enabled together in one marking or not. It has already been motivated why in the field of workflow performance modeling, among other areas, a mixture of both stochastic and fixed durations are necessary. The transition firing delays adopted here can either be zero (immediate), exponentially distributed, deterministic, or may belong to a class of general distributions called *expolynomial*. If no more than one transition with non-exponentially distributed firing delay is enabled in each marking, the underlying stochastic process is *semi-regenerative* and can be directly numerically analyzed.[18] The user-defined performance measures can therefore be computed.

For many models the restriction of not more than one enabled non-exponential transition per marking is violated.[19] Another problem of all analysis methods is the size of the reachability graph. Not only the computational complexity grows, it makes the analysis impossible for some models of realistic size due to memory space restrictions. Discrete-event simulation is still applicable for the performance evaluation in this case. However, other problems arise with the statistical evaluation of the samples and the accuracy of the results.

The evaluation of the model can be used to answer questions like

- How many documents can be processed per week with the modeled organization?
- What is the mean time for a case to be finished?
- How big is the utilization of the resources?
- What are the bottlenecks?
- How much time does a document spend during processing, waiting, or being transported?
- How will the above numbers change if the available staff decreases e.g. due to holidays?

5.4 Analysis of the Application Example

In this section the application example is evaluated. We assume that there are funds available for a total staff of seven employees. The following question is considered: how many employees should be associated with each one of the three offices (file administration, check/decision, damage event handling)?

For the realistic description of times for the process steps, exponentially distributed firing delays have been associated with the normal office tasks, while transitions modeling deadlines and in-house communication fire after a fixed (deterministic) delay. Therefore the structural restriction of not more than one enabled transition with non-exponentially distributed firing delay is violated, forbidding the use of the direct numerical analysis methods. The simulation component of the tool TimeNET is thus used for the performance evaluation of the application example. To enforce a comparably high accuracy of the simulation a confidence interval of 99% and a relative error probability of 3% has been chosen. The simulations have been carried out on a cluster of ten UltraSparc workstations, and each run took typically 49 seconds of total CPU time to finish.

file administration/ check/ damage event handling	1 / 1 / 5	1 / 2 / 4	1 / 3 / 3	1 / 4 / 2	1 / 5 / 1
file administration	99.684 %	100 %	99.764 %	99.284 %	98.684 %
check	94.083 %	74.321 %	71.473 %	67.821 %	64.889 %
damage event handling	99.895 %	99.892 %	99.797 %	99.667 %	100 %
throughput	2.28	2.90	2.40	1.98	1.87
file administration/ check/ damage event handling	2 / 1 / 4	2 / 2 / 3	2 / 3 / 2	2 / 4 / 1	3 / 1 / 3
file administration	91.165 %	91.301 %	91.088 %	90.939 %	78.597 %
check	98.992 %	96.617 %	87.451 %	80.370 %	99.648 %
damage event handling	99.602 %	99.716 %	99.881 %	100 %	99.776 %
throughput	0.68	0.70	0.67	0.64	0.56
file administration/ check/ damage event handling	3 / 2 / 2	3 / 3 / 1	4 / 1 / 2	4 / 2 / 1	5 / 1 / 1
file administration	77.721 %	77.297 %	72.229 %	71.865 %	68.166 %
check	98.067 %	90.447 %	100 %	95.072 %	100 %
damage event handling	99.917 %	100 %	99.670 %	99.878 %	99.752 %
throughput	0.53	0.51	0.39	0.38	0.35

Fig. 10. Simulation results of the performance evaluation

All 15 possible combinations of employee associations have been evaluated. The results are shown in the tables in Figure 10. In the top of each column the number of employees associated to the three offices are given in the following order: file administration, check/decision, and damage event handling. The following three rows show the "utilization" of each office. The numbers show the percentage of time that an employee is busy, provided he is in the office (thus without counting absence times due to holidays etc.). This is important for a bottleneck analysis of the workflow system. The last row gives the most important result, namely the throughput (of cases per time unit) of the organization, that has to be maximized for the application example.

In the cases where the distribution of employees is more or less balanced for the three offices, the utilization results clearly point out that the damage event handling office is the main bottleneck of the system. It is therefore not surprising that a considerable gain in the throughout can be achieved by associating more people to this office. After this has been done, it is advisable to have more employees working in the check/decision office than in the file administration, because this seems to be the second bottleneck. The optimal configuration is 1/2/4. In addition to the shown evaluations one possible configuration (1/1/4) for only six employees has been evaluated, resulting in a throughput of 1.77, which is better than many of the evaluated combinations of seven employees.

6 Tool Support

Modeling and evaluation of complex systems is only feasible with the support of appropriate software tools. The graphical user interface is the basis for the modeling process. Secondly, the algorithms needed for the analysis have to be efficiently implemented and integrated with the user interface.

For the work presented here the software tool TimeNET is used, a tool for the modeling and performability evaluation using stochastic Petri nets. The tool has been designed especially for models with non-exponentially distributed firing delays. Recently, version 3.0 has been released with several extensions of analysis algorithms as well as a new generic graphical user interface (shown in Figure 11).

A general overview of the software package can be found in.[10] One of the recent enhancements is an environment for the modeling and performance evaluation of manufacturing systems based on colored stochastic Petri nets. As work-

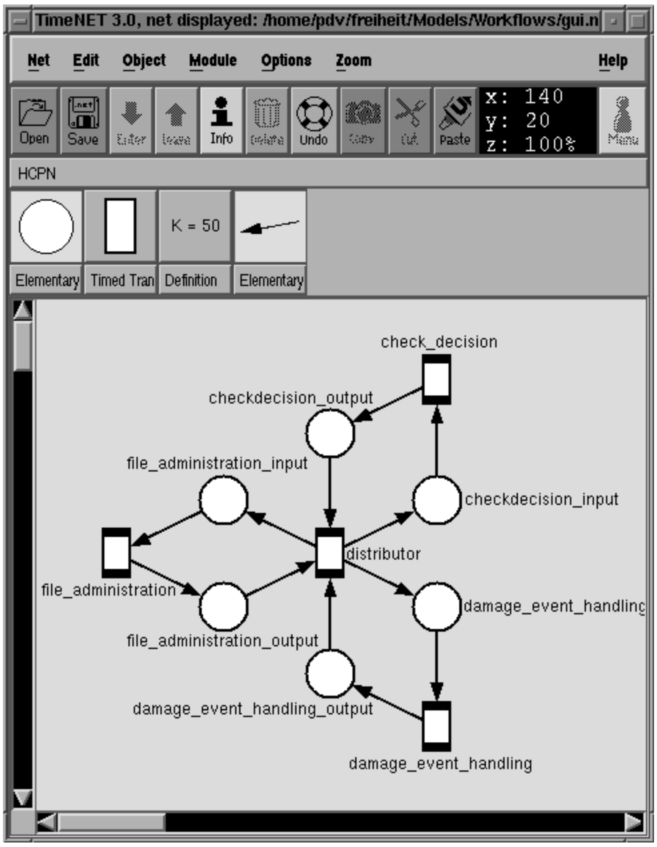

Fig. 11. Sample screenshot of the graphical user interface

flow problems and manufacturing systems share several common characteristics, the tool can be used in this field as well.

7 Conclusion

In this paper a method for the modeling and performance evaluation of workflow systems based on colored stochastic Petri nets is introduced. The approach combines control flow and organizational parts, which are described in clearly separated models. There is no restriction on the number of different business processes of the organization that can be modeled and analyzed together, making the evaluation of their combined effects possible. Deterministic as well as stochastic delays of activities are allowed in the approach and facilitate a more realistic modeling of the processes. The performance of a modeled system can be evaluated by direct numerical analysis or discrete-event simulation. For the application of the described techniques to an application example of medium size the software tool TimeNET has been used.

The paper concentrates on the modeling and simulation of business processes. In the future it should be possible to augment the tool such that run time support including resource monitoring and deadline checking is possible.

References

1. W.M.P. van der Aalst. Three Good Reasons for Using a Petri-net-based Workflow Management System. In S. Narathe and T. Wakayama, editors, *Proc. Int. Working COnference on Information and Process Integration in Enterprises (IPIC'96)*, pages 179–201, Camebridge, Massachussets, November 1996.
2. W. van der Aalst, J. Desel, and A. Oberweis, editors. *Business Process Management: Models, Techniques and Empirical Studies*, volume 1806 of *LNCS*. Springer-Verlag, 2000.
3. J. Dehnert. Four systematic steps towards sound business process models. In H. Weber, H. Ehrig, and W. Reisig, editors, *2nd Int. Colloquium on Petri Net Technologies for Modelling Communication Based Systems*. Fraunhofer Gesellschaft ISST, September 2001.
4. W.M.P. van der Aalst. The Application of Petri Nets to Workflow Management. *The Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.
5. J. Desel and T. Erwin. Modeling, simulation and analysis of business processes. In W. van der Aalst, J. Desel, and A. Oberweis, editors, *Business Process Management: Models, Techniques and Empirical Studies*, volume 1806 of *LNCS*. Springer-Verlag, 2000.
6. C.A. Ellis, K. Keddara, and J. Wainer. Modeling Workflow Dynamic Change Using Timed Hybrid Flow Nets. In W.M.P. van der Aalst, G. De Michelis, and C. A. Ellis, editors, *Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98)*, volume 98/7 of *Computing Science Reports*, pages 109–128. Eindhoven University of Technology, Eindhoven, 1998.
7. V. Atluri N.R. Adam and W-K. Huang. Modeling and analysis of workflows using petri net. *Journal of Intelligent Information System, Special Issue on Workflow and Process Management*, 10(2):131–158, 1998.

8. W.M.P. van der Aalst. Interval Timed Coloured Petri Nets and their Analysis. In M. Ajmone Marsan, editor, *Application and Theory of Petri Nets 1993*, volume 691, pages 453–472. Springer-Verlag, Berlin, 1993.
9. J. Eder, E. Panagos, H. Pozewaunig, and M. Rabinovich. Time Management in Workflow Systems. In W. Abramowicz and M.E. Orlowska, editors, *BIS'99 3rd International Conference on Business Information Systems*, pages 265–280, London Berlin Heidelberg, jun 1999. Springer Verlag.
10. A. Zimmermann, R. German, J. Freiheit, and G. Hommel. TimeNET 3.0 tool description. In *Petri Nets and Performance Models 1999, Tool Descriptions*, Zaragoza, Spain, 1999.
11. W. Reisig. *A Primer in Petri Net Design*. Springer-Verlag, 1992.
12. T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, 1989.
13. K. Jensen. *Coloured Petri nets, basic concepts, analysis methods and practical use*, volume 1. Springer-Verlag, 1992.
14. M.K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Trans. Computers*, C-31(9):913–917, 1982.
15. M. Ajmone Marsan, G. Conte, and G. Balbo. A class of generalized Petri nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comput. Sys.*, 2(2):93–122, 1984.
16. M. Ajmone Marsan and G. Chiola. On Petri nets with deterministic and exponential transition firing time. In *Advances in Petri Nets*, volume 266 of *Lecture Notes in Computer Science*, pages 132–145. Springer-Verlag, 1987.
17. A. Zimmermann, S. Bode, and G. Hommel. Performance and dependability evaluation of manufacturing systems using petri nets. In *1st Workshop on Manufacturing Systems and Petri Nets, 17th Int. Conf. on Application and Theory of Petri Nets*, pages 235–250, Osaka, Japan, 1996.
18. G. Ciardo, R. German, and C. Lindemann. A characterization of the stochastic process underlying a stochastic petri net. *IEEE Transactions on Software Engineering*, 20:506–515, 1994.
19. R. German. *Performance Analysis of Communication Systems: Modeling with Non-Markovian Stochastic Petri Nets*. John Wiley and Sons, 2000.