ulm university universität
uulm

Time Patterns for Process-aware Information Systems: A Pattern-based Analysis

Barbara Weber, Andreas Lanz, Manfred Reichert

Ulmer Informatik-Berichte

Nr. 2009-05
März 2009

Time Patterns for Process-aware Information Systems: A Pattern-based Analysis

Barbara Weber[1], Andreas Lanz[2], and Manfred Reichert[2]

[1]Quality Engineering Research Group, University of Innsbruck, Austria
Barbara.Weber@uibk.ac.at
[2]Institute of Databases and Information Systems, Ulm University, Germany
{Andreas.Lanz,Manfred.Reichert}@uni-ulm.de

Abstract. Formal specification and operational support of time constraints constitute fundamental challenges for any enterprise information system. Although temporal constraints play an important role in the context of long-running business processes, time support is very limited in existing process management systems. By contrast, different kinds of planning tools (e.g., calendar systems and project management tools) provide more sophisticated facilities for handling task-related time constraints, but lack an operational support for business processes. This paper presents a set of 10 time patterns to foster the systematic comparison of these different technologies in respect to time management. The proposed patterns are all based on empirical evidence from several large case studies. In addition, we provide an in-depth evaluation of selected process management systems, calendar systems and project management tools based on the suggested patterns. The presented work will not only facilitate comparison of these different technologies in respect to their support of time constraints, but also make evident that their integration offers promising perspectives in respect to time support for long-running business processes.

1 Introduction

Formal specification and operational support of time constraints constitute fundamental challenges for any enterprise information system. Although temporal constraints play an important role in the context of long-running business processes (e.g., patient treatment, automotive engineering and flight planning) [1, 2, 3, 4], time support is limited in existing process management systems [1, 5]. By contrast, different kinds of planning tools (e.g., calendar systems and project management tools) provide sophisticated facilities for handling time constraints (e.g., periodic activities), but miss an operational support for business processes. So far, there is a lack of methods for systematically assessing and comparing the time capabilities provided by these different process support technologies (denoted as *process-aware information systems* (PAISs) in the following).

To make PAISs better comparable and to facilitate the selection of appropriate PAIS-enabling technologies, workflow patterns have been introduced [6, 7, 8, 9]. Respective patterns provide means for analyzing the expressiveness of process modeling approaches in respect to different process perspectives. In particular, proposed workflows patterns cover control flow [6], data flow [7], resources [8], exceptions [10], and process change [9, 11]. However, a framework for systematically evaluating existing PAISs in respect to their ability to deal with time aspects is missing and is picked up by this paper. Our contributions are twofold.

1.) We suggest 10 *time patterns* to foster the comparison of existing PAISs with respect to their ability to deal with time aspects. The proposed time patterns complement existing workflow patterns and were systematically identified by analyzing a large collection of process models in healthcare, automotive engineering, aviation industry, and other domains.
2.) We provide an *in-depth evaluation* of selected approaches from both industry and academia based on the proposed time patterns. The evaluation does not only consider process management systems, but also calendar systems and project planning tools in which time aspects play an important role.

Our pattern-based analysis shows that these different technologies all provide support for time aspects. The presented work will not only facilitate their comparison in respect to the support of time constraints, but also foster the selection of appropriate time components when designing PAISs. Moreover, our work makes evident that their integration offers promising perspectives in respect to more sophisticated time support for long-running business processes, i.e., knowing the commonalities and differences will be a first step to integrate these technologies (e.g., process management and calendar systems).

Section 2 summarizes background information. Section 3 presents the research method employed for identifying the time patterns. Section 4 describes 10 time patterns sub-dividing them into 4 categories. Section 5 evaluates different approaches from academia as well as industry (detailed result of this evaluation can be found in Appendix A). We present related work in Section 6 and conclude with a summary and outlook in Section 7.

2 Background Information

In this section we describe basic concepts and notions used in this paper.

A process management system is a specific type of information system which provides process support functions and separates process logic from application code. For each business process to be supported, a *process type* represented by a *process schema* has to be defined (cf. Fig. 1). In the following, a process schema corresponds to a directed graph, which comprises a set of *nodes* – representing *activities* or *control connectors* (e.g., XOR/AND-Split, XOR/AND-Join) – and a set of *control edges* between them. The latter specify precedence relations. We further use the notion of *activity set* to refer to a subset of the activities

of a process schema. Its elements are not required to be part of a sequence block, but can also belong to different parallel branches. During run-time *process instances* are created and executed according to a predefined process schema S. *Activity instances*, in turn, represent a single process step of a particular process instance. Activities which can be executed more than once (e.g., being executed concurrently or sequentially) are referred to as *multi-instance activities*. The patterns introduced in the following can be applied to all these granularities (i.e., process schema, activity, activity set, activity instance, and process instance). We use the term *process element* as umbrella for all these concepts.

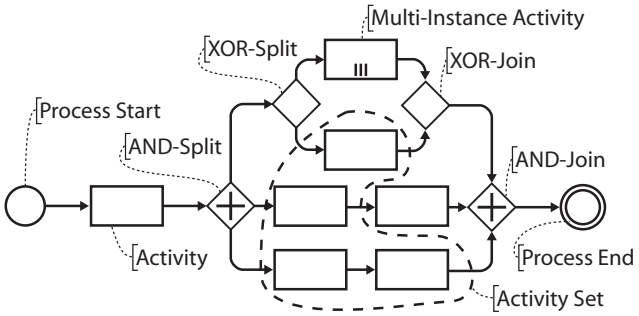


Fig. 1. Core Concepts

3 Research Method

The overall goal of this paper is to complement existing workflow patterns with a set of time patterns suitable to assess how effectively PAISs can deal with time. As motivated in the introduction, adequate modeling and management of temporal constrains will be a key ability of future PAISs, particularly in connection with the support of long-running processes.

3.1 Pattern Identification

We describe the selection criteria for our time patterns, the data sources they are based on, and the procedure we applied for pattern identification.

Selection Criteria. This paper considers patterns covering relevant temporal aspects for the modeling and control of business processes and business activities respectively. Our focus is on a high coverage of real-world scenarios, and not on specific time features of a PAIS like verification of time constraints [2, 5, 1, 3], escalation management [12], or scheduling support [13, 14]. Finally, we also exclude history-based time considerations (e.g., analyzing execution traces).

Sources of Data and Data Collection. As sources for our patterns we consider results of comprehensive case studies we performed in different domains, including healthcare, automotive engineering, aviation industry and others. One of our major data sources is a large BPM project at a Women's Hospital in which

all core processes of the hospital were analyzed and modeled (either using ARIS or Bonapart as process modeling tool) (e.g., [15, 16, 17, 18, 19]). Further, selected processes were implemented using the WorkParty workflow system. As part of this project time aspects were elicitated and documented in detail. In total we consider 98 process models covering both administrative processes (e.g., patient admission or order handling) and medical treatment processes (e.g., in-patient chemotherapy and ovarian cancer surgery).

As our second major data source we use process models from the automotive domain. We consider a case study on electronic change management (ECM) [20] and process models described in [21]. The models related to ECM have been published by the German Association of the Automotive Industry (VDA) [20]. The process models described in [21], in turn, refer to car repair and maintenance in garages, in-house change management, and product development. With several hundred activities the product planning development is the most complex one we consider. In total, our material from this project consists of 59 process models.

As third data source serves a case study we conducted with an on-demand air service. As part of this project we analyzed and documented the flight planning and post flight phase. As the aviation industry is highly regulated compliance with existing standards and regulations in addition to company policies is essential (e.g., minimum standards for Flight Time Limitations, or company internal duty and rest time regulations). Many of these regulations contain time constraints which have to be obeyed.

Our fourth data source constitute healthcare processes we encountered when working at a large Medical University Hospital. This includes about 60 different processes, related to the diagnostic and therapeutic procedures in the field of internal medicine (e.g., examinations in medical units like radiology, gastroenterology, clinical chemistry). One of the authors works on the developement of a large Scheduling system for outpatients.

Pattern Identification Procedure. To ground our patterns on a solid basis we first create a list of candidate patterns. For generating this initial list we conduct a detailed literature review and rely on our experience with PAIS-enabling technologies. Next we thoroughly analyze the above mentioned material to find empirical evidence for our time patterns and - if necessary - extend the candidate list of patterns. As a pattern is defined as reusable solution to a commonly occurring problem we require each of our time patterns to be observed at least three times in different models of our process samples. Therefore, only those patterns, for which enough empirical evidence exists, are included in the final list of patterns, which is presented in Section 4.

4 Time Patterns

As result of our analysis we have identified 10 different patterns which we divide into 4 distinct categories (cf. Fig. 2). All these time patterns constitute solutions for realizing commonly occurring time aspects in PAISs. Pattern Category

I (*Durations and Time Lags*) provides support for expressing durations of process elements (e.g., activities) as well as time lags between events or activities. Pattern Category II (*Restrictions of Process Execution Points*) allows specifying constraints regarding possible execution points of process elements (e.g., activity deadline). Category III (*Variability*) provides support for time based variability (e.g., control-flow varies depending on time context). Finally, Category IV (*Reoccurring Process Elements*) comprises patterns for supporting reoccurring process elements (e.g., periodicity and cyclic flows).

Pattern Catalogue
Category I: Durations and Time Lags
TP1: Time Lags between Events
TP2: Durations
TP3: Time Lags between Activities
Category II: Restrictions of Process Execution Points
TP4: Fixed Date Elements
TP5: Schedule Restricted Elements
TP6: Time Based Restrictions
TP7: Validity Period
Category III: Variability
TP8: Time Dependent Variability
Category IV: Reoccurring Process Elements
TP9: Cyclic Elements
TP10: Periodicity

General Design Choices
A.) When do time parameters have to be set? a.) At build-time (i.e., during process modeling) b.) At instantiation time (i.e., when a process instance is instantiated) c.) At run-time (i.e., during process execution)
B.) What granularities are supported by the reference system? a.) Basic (i.e., years, months, weeks, days, hours, minutes, seconds) b.) System-defined (e.g., business days) c.) User-defined (e.g., Wednesday afternoon)

Fig. 2. Pattern Catalogue (left) and General Design Choices (right)

Fig. 2 gives an overview of the 10 time patterns, which are described in detail in the following. For each pattern we provide a name, a brief description including a visualization, synonyms (if applicable), a couple of design choices, remarks regarding its implementation, illustrating examples, and a reference to related patterns. In particular, *design choices* allow for parametrizing time patterns keeping the number of distinct patterns manageable. Design choices not only relevant for a particular pattern, but for all patterns, are described only once for the entire set. Typically, existing approaches only support a subset of the design choices in the context of a particular pattern. We denote the combination of design choices supported by a particular approach as *pattern variant*.

Fig. 2 also describes two general design choices, which are valid for all 10 time patterns and which can be used for their parametrization. Additional design choices, only relevant in the context of a specific time pattern, are provided with the respective patterns description. The design choices listed in Fig. 2 are shortly described in the following. The time parameters of a time pattern can be set at built-time, at instantiation time, or at run-time (Design Choice A). The time parameters of each time pattern can further have different granularities (depending on what granularities are supported by the time reference system). Typical granularities are years, months, weeks, days, hours, minutes and seconds, but also system-defined granularities (e.g., business days) or user-defined ones (e.g., Wednesday afternoon) (Design Choice B).

Pattern Category I: Durations and Time Lags. This first pattern category comprises 3 basic time patterns determining durations of process elements as well as time lags between them.

Pattern TP1 (*Time Lags between Events*) allows specifying time lags between two discrete events. Events may, for example, correspond to the start or completion of an activity or process instance, cover the reaching of a milestone within a process instance, or correspond to the occurrence of a heart stroke (cf. Fig. 3). In addition to the design choices from Fig. 2, TP1 has two pattern-specific ones. Design Choice C describes the kind of time lags that can be expressed, i.e., minimum / maximum time lags, time intervals and average values. Design Choice D, in turn, specifies whether time lags are described quantitatively (e.g., 4 business days) or qualitatively (e.g., before, during) [22].

Pattern TP2 (*Durations*) allows specifying the durations of process elements (cf. Fig. 4). Design Choice E describes whether the duration constraint applies to a single activity, a set of activities, a process, or a set of process instances. Design Choice F, in turn, describes whether the specified durations are minimum values, maximum values, time intervals, or average values. TP2 can be implemented based on TP1, i.e., a duration corresponds to the time lag between the start of a process element and its completion. When compared to TP1, TP2 allows for a higher level of abstraction. However, TP1 provides more generic support for expressing time lags and is not restricted to start and completion events of activities.

Pattern TP3 (*Time Lags between Activities*) supports the definition of time lags between two activities (cf. Fig. 5). Design Choice G describes whether the specified durations are minimum values, maximum values, time intervals, or average values. Design Choice H, in turn, describes whether the time lags describe a start-start relation (e.g., between the start event of two different activities), a start-end relation, an end-start relation, or an end-end relation. Design Choice I specifies whether the time lags are described quantitatively or qualitatively. Finally, Design Choice J defines whether time lags can only be expressed between two directly succeeding activities or between two arbitrary activities. Like TP2, TP3 can be implemented using TP1, but supports a higher level of abstraction as typically found in project management tools and netplans (e.g., CPM, MPM) respectively.

Pattern Category II: Restrictions of Process Execution Points. This pattern category comprises four patterns for restricting the execution points of process elements.

Pattern TP4 (*Fixed Date Element*) provides means for specifying deadlines. Deadlines can be defined for either an activity instance, for multiple instances of a single activity, or for a process instance (Design Choice K). For a particular process element it can be fixed whether it has to be executed at, completed before or started after a particular date (Design Choice L). In many cases fixed date elements also determine the latest or earliest start / completion time of preceding / succeeding activities.

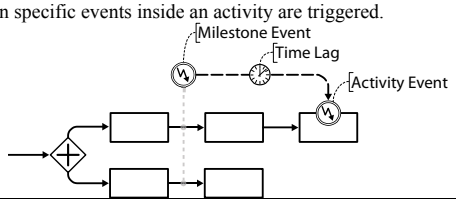
TP1: Time Lags between Events	
Description	There are time lags between any two discrete events. For example, events occur when instantiating or completing a process instance, when reaching a milestone in a process instance, or when specific events inside an activity are triggered. 
Design Choices	C.) What kinds of time lags are used? a.) Minimum value, b.) Maximum value, c.) Time interval [min … max] and d.) Average value
	D.) How are time lags expressed? a.) Quantitatively (e.g., 5 hours, 3 business days) b.) Qualitatively (e.g., before, after)
Remark	Time lags are often required to comply with existing rules and regulations. Further, they may or may not have binding character.
Example	• *Maximum time lags* in an electronic change management process between sending a request for comments (by the partners affected by a change) and getting a response (Design Choices C[b] D[a]). • Aircraft maintenance intervals for conducting an "A check" are *1 month or 500 flight hours*, i.e., the time lag between two "A checks" must not be longer than 1 month and not exceed 500 flight hours (Design Choices C[b] D[a]). • The time lag between two heavy maintenance visits of an aircraft is *4-5 years* (Design Choices C[c] D[a]). • 6 months before contract expiry, customers are informed by the leasing company (Design Choices C[b] D[a]). • The *average time lag* between delivery of all parts (milestone) and the assembly of the car's chassis (milestone) should be approximately *2 hours* (e.g. just-in-time production) (Design Choices C[d] D[a]).
Related Patterns	TP2 – Durations TP3 – Time Lags between Activities

Fig. 3. TP1 - Time Lags between Events

Pattern TP5 (*Schedule Restricted Element*) allows restricting the execution of a particular element by a schedule. The schedule itself is known at built-time, whereas the concrete dates are specified either at instantiation or run-time. Schedules can be defined for a single activity or an entire process (Design Choice M). The schedule may comprise several discrete points in time, but also one or more time frames (Design Choice N). Finally, Design Choice O specifies whether or not exceptions to the schedule may be explicitly defined.

Pattern TP6 (*Time Based Restrictions*) provides support for restricting the number of times a particular process element can be executed within a predefined time frame. Design Choice P describes to which process element the pattern may be applied (e.g., activities instances of a single process instance or of multiple process instances). In addition, TP6 can either restrict the number of concurrent executions or the overall number of executions per time period (Design Choice Q).

Finally, Pattern TP7 (*Validity Period*) allows restricting the lifetime of a process element to a given validity period. Validity periods can be specified for both

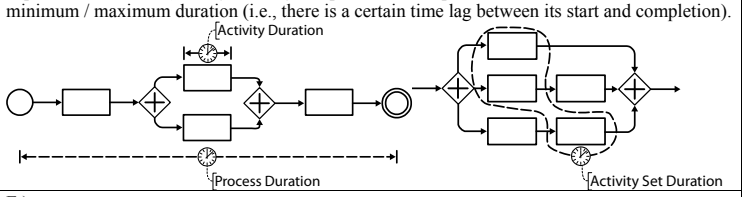
TP2: Durations	
Description	A particular element (activity, activity set, process, set of process instances) has a certain minimum / maximum duration (i.e., there is a certain time lag between its start and completion). 
Design Choices	E.) To which process elements may the pattern be applied? a.) Single activity b.) Set of activities c.) Process model (i.e., duration constraint must hold for all instances of the process) d.) Set of Process Instances (i.e., duration of a set of process instances)
	F.) What kinds of durations are used? a.) Minimum value, b.) Maximum value, c.) Time interval [min … max] and d.) Average value
Remark	Durations may or may not have binding character. Durations are the result of both waiting and processing times. Upper or lower bounds for durations are often determined by external benchmarks (e.g., regulations, policies or QoS agreements).
Example	***Durations of single activities:*** • Oral exams take *at least 30 minutes* (Design Choices E[a] F[a]) • Each round of a box fight *lasts for 2 minutes* (Design Choices E[a] F[b]) • The assembly of a new engine must *not* take *longer than 30 minutes* (task work) (Design Choices E[a] F[b]) • Depending on its severity ovarian cancer surgeries take *1 to 10 hours* (Design Choices E[a] F[c]). ***Durations of activity set*** • Each round in a chess tournament can take *at most 30 minutes*; a round consists of several iterations of "black moves" and "white moves" activities. Each player can *spent 15 minutes in total* per round for all her moves (duration Design Choices E[b] F[b]) • The duty hours of a pilot / year (i.e., sum of flight duty periods) must *not exceed 2000 hours* (Design Choices E[b] F[b]). ***Durations of process*** • From applying for a new passport until its receipt it usually takes about *5 working days* (Design Choices E[c] F[d]). • Maintenance issues need to be resolved *within 1hr* (Design Choices E[c] F[b]) • The contract duration of most leasing processes is *3 years* (Design Choices E[c] F[d]). ***Duration of process instance set*** • Processing 1000 requests must not take *longer than 1 second* (Design Choices E[d] F[b])
Related Patterns	TP1 – Time Lags between Events – TP2 can be implemented based on TP1 TP3 – Time Lags between Activities

Fig. 4. TP2 - Durations

activities and processes (Design Choice R). Design Choice S allows specifying the earliest / latest start date or the latest completion date for the respective process element.

Pattern Category III: Variability. This pattern category comprises Pattern TP8 (*Time Dependent Variability*) which allows varying the control-flow depending on the execution time or time lags between activities or events (Design Choice T).

Pattern Category IV: Reoccurring Process Elements. This category comprises two fundamental patterns to express cyclic elements and periodicity.

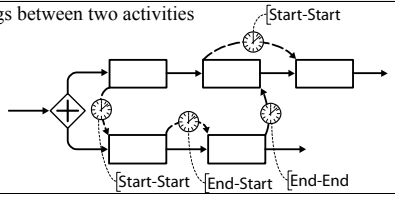
TP3: Time Lags between Activities	
Description	There are time lags between two activities 
Synonyms	Often referred to as “Upper and Lower Bound Constraints”, “Inter-Task Constraints” or “Temporal Relations".
Design Choices	G.) What time relations are supported? a.) Time lag between the start of two activities (i.e., Start-Start) b.) Time lag between the start of the first and the completion of the second activity (i.e., Start-End) c.) Time Lag between the completion of the first and the start of the second activity (i.e., End-Start) d.) Time Lag between the completion of two activities (i.e., End-End)
	H.) What kinds of time lags are used? a.) Minimum value, b.) Maximum value, c.) Time interval [min … max] and d.) Average value
	I.) How are time lags expressed? a.) Quantitatively (e.g., 5 hours, 3 business days) b.) Qualitatively (e.g., before, after)
	J.) Time lags can be expressed a.) Only between succeeding activities b.) Between any two activities
Remark	Time lags are often required to comply with existing rules and regulations; Like durations time lags may or may not have binding character.
Example	• The *maximum time lag* between discharge of a patient from a hospital and sending out the discharge letter to the general practitioner of the patient should be 2 weeks (Design Choices G[d] H[b] I[a]) • Patients must not eat *at least 12 hours* before the surgery takes place. The latest point in time where the patient can have a meal is determined by the date of the surgery (Design Choices G[c] H[b] I[a]) • A contrast medium has to be administered *2 to 3 hours before* a radiological examination. The interval in which the contrast medium should be administered depends on the examination date (Design Choices G[a] H[c] I[a]) • The time lag between registering a master thesis and submitting it must *not exceed 6 months* (Design Choices G[a] H[b] I[a]) • Activity “Sell foods and drinks” *starts with the first round of the chess tournament and ends with the flower ceremony* (Design Choices G[a;d] H[b] I[b] J[a;b])
Related Patterns	TP1 – Time Lags between Events; TP3 can be implemented based on pattern TP1 TP2 – Durations

Fig. 5. TP3 - Time Lags between Activities

Pattern TP9 (*Cyclic Elements*) allows specifying cyclic elements which are performed iteratively considering time lags between cycles. Design Choice U specifies whether time lags between cycles are fixed (i.e., have always same length), are fuzzy (e.g., 2 to 3 hours), or may vary from iteration to iteration. Design Choice V describes whether the number of cycles is determined explicitly, calculated based on time lags and end dates, or depending on an exit condition.

Pattern TP10 (*Periodicity*) allows specifying periodically reoccurring process elements according to an explicit *periodicity* rule. In contrast to TP9 the emphasis of TP10 is on possible execution dates of the reoccurring element and not on the time lags between the iterations. Design Choice W describes whether the

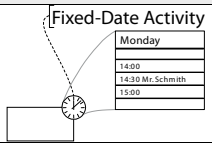
TP4: Fixed Date Elements	
Description	A particular element (i.e., activity instance or process instance) has to be executed at /completed before / started after a particular date. 
Synonyms	Often referred to as "Deadline".
Design Choices	K.) To which process elements can the pattern be applied? a.) An activity instance b.) Multiple instances of a single activity c.) A process instance
	L.) What type of date is specified? a.) Execution date b.) Latest completion date c.) Earliest start date
Remark	Fixed Date Elements often determine the latest or earliest start / completion time of preceding / succeeding activities, e.g., when flying from Munich to Amsterdam at 6:05, check-in must be no later than 5:20 (at least 45 minutes before departure). If this deadline cannot be made, the flight will be missed (element or even process becomes obsolete).
Example	• Assume that software is released every two weeks on Friday evening; Thus, the deadline for changes (except bug fixes) is the *day before the release date* (time error might lead to delay or have no effect) (Design Choices K[a] L[b]). • To perform chemotherapy the physician has to inform the pharmacy about the dosage of the cytostatic drug *until 11:00*. If the deadline is missed the pharmacy checks back by phone for the exact dosage (escalation mechanism) (Design Choices K[a] L[b]). • A patient has an appointment for an examination *Monday at 10:00,* but due to a full schedule of the physician it may well be that the patient has to wait until the examination starts (i.e., earliest possible execution point is given) (Design Choices K[a] L[c]). • To handle a change request in the automotive domain (e.g., in the design of the side door) a statement of all affected partners has to be requested (i.e., multiple activity instances with a priori run-time knowledge). All statements have to be available *before a particular deadline* (Design Choices K[b] L[b]).
Related Patterns	TP5 – Schedule Restricted Elements; Fixed Date Elements are often also schedule restricted elements.

Fig. 6. TP4 - Fixed Date Elements

periodicity rule may contain one or more dates. In addition, Design Choice X specifies whether the number of cycles is determined explicitly, calculated based on time lags and end dates, or depending on an exit condition. Finally, Design Choice Y describes whether exceptions to the periodicity rule can be specified.

5 Evaluation

In the following we describe the evaluation of selected approaches from academia and industry regarding their support for time patterns. Section 5.1 describes our evaluation methodology, while Section 5.2 shows our evaluation results.

5.1 Evaluation Methodology

This section sketches the methodology employed for conducting our evaluation. In particular, we describe evaluation goal, evaluation objects, evaluation criteria, evaluation metrics, and the evaluation procedure.

Definition of Evaluation Goal. The goal of our evaluation is to measure how well current PAISs cope with time aspects.

TP5: Schedule Restricted Elements	
Description	The execution of a particular element (i.e., activity, or process) is restricted by a schedule. The schedule (i.e., structure) is known at type level, while the concrete date is specified at instance level.
Design Choices	M.) To which process elements can the pattern be applied? a.) Single activity b.) One process
	N.) The execution of the element can be bound to a.) Several discrete points in time (execution is only possible every full hour) or b.) One or more time frames (execution is only possible from 09:00 to 12:00 and from 13:00 to 16:00)
	O.) Can exceptions to the schedule be specified (e.g., every year except leap years)? a.) Yes b.) No
Remark	The schedule attached to a schedule restricted element provides restrictions on when the element can be executed. In particular for rather restricted schedules even small delays in process execution can be critical (if schedule restricted elements being on a critical path are affected by the delay or the path becomes critical due to the delays). Assume that the ferry between Happy Valley Goose Bay (Labrador) and Cartwright (Labrador) operates only once a week. If the respective ferry is missed (by a couple of minutes), this will lead to a delay of 1 week of the respective activity. To avoid delays schedule restricted elements often become fixed date elements as soon as the execution time of the element gets fixed (e.g., when a particular ferry for travelling from Happy Valley Goose Bay to Cartwright is chosen).
Example	• Between Munich and Amsterdam there are flights *at 6:05, 10:30, 12:25, 17:35 and 20:40* (Design Choice B[a] M[a] N[a] O[b]). • Opening hours of the dermatological clinic are *MO – FR 8:00 – 17:00 except for public holidays*. Dermatological examinations can only be scheduled within this time frame (Design Choices B[a] M[a] N[b] O[a]). • An information letter is sent by the leasing company to each customer *within the first two weeks of each year* (Design Choices B[a] M[a] N[b] O[b]) • Particular examinations (e.g., punctures) are only conducted *once a week* (e.g., Wednesday afternoon) (Design Choices B[b] M[a] N[b] O[b]) • Comprehensive lab tests in a hospital can only be done *from MO – FR 8:00 – 17:00* (Design Choices B[a] M[a] N[b] O[b])
Related Patterns	TP4 – Fixed Date Elements (often schedule restricted elements) TP6 – Time Based Restrictions (like schedule based restrictions constrain possible execution points for an element) TP7 – Validity Period

Fig. 7. TP5 - Schedule Restricted Element

Selection of Evaluation Objects. As evaluation objects we choose process management systems, calendar systems, and project planning tools from both academia and industry. In terms of academic approaches our evaluation considers the proposals made by Eder [3], Bettini [5], and Combi [1]. As samples for commercial systems our evaluation includes the process management systems MQSeries Workflow and Tibco iProcess Modeller, for which we have hands-on experience as well as running installations in our labs. Further, we include the widely used calendar systems Outlook, Google Calendar and Lightning, and the well-known project management tool MS Project. Finally, with BPMN our evaluation comprises a commonly used process modeling language.

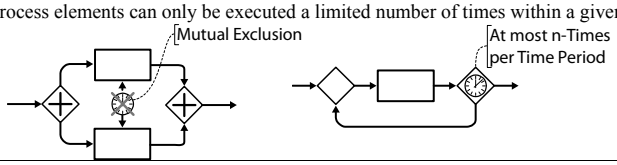
TP6: Time Based Restrictions	
Description	Particular process elements can only be executed a limited number of times within a given timeframe. 
Synonyms	Design Choice Q[a] is often referred to as "Mutual Exclusion"
Design Choices	P.) To which process elements can the pattern be applied? a.) Instances of a single activity or group of activities within same process instance b.) Instances of a single activity or group of activities within different process instances (potentially sharing some common characteristics) c.) Instances of a process or group of processes
	Q.) What kind of restrictions can be expressed? a.) Number of concurrent executions (at same time / with overlapping time frames) b.) Number of executions per time period
Remark	Time Based Restrictions are often used to express the influence of resource restrictions / resource shortage onto the process execution.
Examples	• Two invasive examinations must *not* be performed *on the same day* (Design Choices A[a] P[b] Q[b]). • Several examinations for a particular patient are performed within a limited timeframe; Thereby, it has to be ensured that the patient is *not x-rayed several times* (Design Choices A[a] P[b] Q[b]). • Examinations for a particular patient have to be scheduled sequentially (Design Choices A[a] P[b] Q[a]). Two X-ray activities (for different patients) *cannot be executed at the same time*, as the X-ray machine cannot be shared (Design Choices A[a] P[b] Q[a]). • For USD 19.90 *10* different online books can be read *per month*. If the book tokens are consumed no more books can be read in the current month. At the beginning of next month the book tokens get renewed (Design Choices A[a] P[a] Q[b]). • During your stay at a wellness hotel you can select *one treatment* (free of charge) *per day* (Design Choices A[a] P[a] Q[b]). • The test version of a particular BPM suite only support the execution of *at most 25 concurrent process instances* (Design Choices A[a] P[c] Q[a]).
Related Patterns	TP5 – Schedule Restricted Elements; While the execution point of a schedule restricted element is constraint by a schedule, time based restrictions constrain the amount of activity instances / time period.

Fig. 8. TP6 - Time Based Restrictions

Definition of Evaluation Criteria and Metrics. Evaluation criteria are the 10 change patterns described in Section 4. We measure the ability of a PAIS to deal with time aspects as the degree of support for the described evaluation criteria. For each evaluation criterion we differentiate between *supported*, *partially supported*, *not supported*, and *not specified*. If an evaluation object provides support for a particular criterion the supported design choices are listed. If a particular evaluation object is only partially supported this is additionally labeled with "*". No support is labeled with "-" and *not specified* with "?". Assume that an evaluation object supports Pattern TP4 with Design Choices K and L. Further assume that for Design Choice K Option a is partially supported and for Design Choice L Option b is supported. This would result in the String "K[a*], L[b]" (e.g., Eder et al. in Fig. 13).

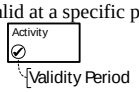
TP7: Validity Period	
Description	A particular activity or process can only be executed within a particular validity period, i.e., its lifetime is restricted to the validity period. Activities or processes can only be instantiated within the validity period. In general, different versions of an activity or process may exist, but only one is valid at a specific point in time. 
Design Choices	R.) To which process elements can the pattern be applied? a.) A single activity b.) A single process
	S.) What type of date is specified? a.) Earliest starting date b.) Latest starting date c.) Latest completion date
Remark	Validity dates are particularly relevant in the context of process evolution, to restrict the remaining lifetime of an obsolete process and to time the rollout of the replacement process.
Example	• *Starting from Jan 1st* patients need to be informed about any risks before the actual treatment takes place (Design Choice R[b] S[a]). • *From next week on* the new service version should get life (Design Choice R[a] S[a]). • *In the promotion period* (lasting from Monday to Friday next week) special conditions apply. Within this period a different version of the price calculation activity should be applied (Design Choice R[a] S[a;b]). • Due to a changed law, process A may only be used *until January 1st*. After this date no new process instances can be instantiated based on A, but process B has to be used instead (Design Choice R[b] S[b]).
Related Patterns	TP5 – Schedule Restricted Elements TP8 – Time Dependent Variability

Fig. 9. TP7 - Validity Period

Analyzing the Evaluation Objects along the Evaluation Criteria. For the academic approaches we base our evaluation on a comprehensive literature study. Regarding commercial systems, support for time patterns was determined based on the installations in our lab and on our hands-on experience with respective systems. A summary of our evaluation results is given in Fig. 13. An in-depth description of each of the evaluated approaches including a detailed description of all supported design choices can be found in Appendix A. Note that this evaluation only considers time patterns. Time features like verification of time constraints, escalation mechanisms or scheduling support are outside of the scope of this paper.

5.2 Evaluation Results

Fig. 13 shows which time patterns are supported by our evaluation objects. Calendar systems like MS Outlook, Google Calendar and Lightening provide good support for Pattern TP4 (*Fixed Date Element*) and Pattern TP10 (*Periodicity*), while limited support is provided for specifying business rules and regulations (i.e., patterns TP1, TP2, TP3 and TP6). In addition to the support provided by calendar systems, project management tools like MS Project provide some

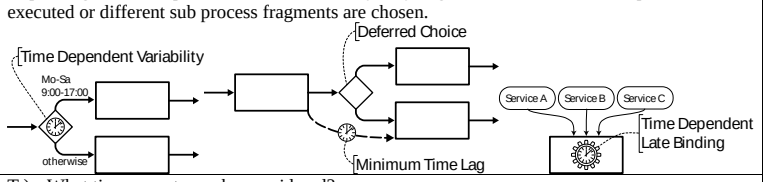
TP8: Time Dependent Variability	
Description	Depending on time aspects the control flow may vary; e.g., different branches of a process are executed or different sub process fragments are chosen. 
Design Choices	T.) What time aspects can be considered? a.) Execution time of an activity instance / process b.) Time lags between activities / events
Remark	Time dependent variability can be achieved in different ways. The simplest approach is to explicitly capture the required variability in the process model through enumerating all different options. Alternatively, techniques like late binding can be used to select appropriate activity implementations during run-time dependent on the time. Finally, time dependent variability can also be achieved by using the Deferred Choice Pattern in combination with triggers (e.g., if no offer is received within 7 days another one will be requested).
Example	• Samples which are collected between 18 and 20 o'clock and which are sent to the Department of Clinical Chemistry, need to be marked as express requests in the request form. Outside the opening hours of the clinic only emergency cases are treated (Design Choice T[a]). • When issuing a passport the processing usually takes 4-6 weeks. If the person needs the passport earlier than 4 weeks an interim passport can be issued (Design Choice T[a]). • When visiting the cinema before 5pm or during the holiday period tickets are cheaper (Design Choice T[a]). • Between 7am and 6pm the lab can analyze all parameters of a sample, while during night only limited services are provided and thus only a limited set of (critical) parameters can be analyzed (Design Choice T[a]). • Patients admitted in the hospital between 6pm and 8am are always assigned to the emergency unit (for the first night); afterwards they are transferred to a normal ward (Design Choice T[a]). Between 8am and 6pm, in turn, patients are directly admitted by the ward. • If no offer is received 7 days after having sent the request another request is sent (Design Choice T[b]). • When ordering goods usually the normal delivery option is selected, however, if the goods are urgently needed express delivery can be selected (Design Choice T[b]).
Related Patterns	TP7 – Validity Dates

Fig. 10. TP8 - Time Dependent Variability

support for specifying business rules and regulations. However, project management systems lack operational support for multiple concurrently executed process instances. BPMN as a representative of a process modeling language provides limited support for time aspects. The support for time constraints in commercial workflow management systems is even more limited and restricted to the definition of maximum execution durations. Academic approaches are comparably more expressive and provide good support for specifying business rules and regulations. However, except for the proposal of Combi et al. [1] the evaluated approaches do not consider any loops resulting in missing support for patterns of Category IV (*Reoccurring Process Elements*). Interestingly, despite their relevance for real world applications support for Pattern TP6 (*Time Based Restrictions*) and Pattern TP7 (*Validity Periods*) is missing in almost all evaluation objects.

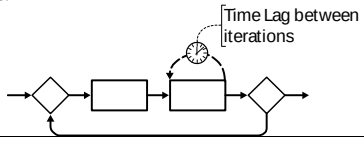
TP9: Cyclic Elements	
Description	A particular activity, activity set, or process shall be performed iteratively considering time lags between the cycles. 
Design Choices	U.) Time lag between cycles is a.) Fixed (e.g., 3 hours) b.) Fuzzy (e.g., 2-3 hours) c.) Can vary
	V.) Number of cycles a.) Fixed / dynamic number of iterations b.) Depends on time lag and end date c.) Depends on exit condition
Example	• Administer 50 to 75 mg in equally divided doses *every 12 hrs for 5 subsequent days* (Design Choices U[a] V[a]). • Administer 1 ml *every 2 to 3 hours until symptoms improve* (Design Choices U[b] V[c]). • Maintenance activities for a particular aircraft have to be performed *after every N flight hours* (Design Choices U[a] V[c]). • Maintenance Aircraft "C Checks" are performed *every 12-18 months or 2500 flight hours* (Design Choices U[a] V[c]).
Related Patterns	TP10 – Periodicity

Fig. 11. TP9 - Cyclic Elements

6 Related Work

Patterns were first used by Christopher Alexander [23] to describe solutions to recurring problems and best practices in architectural design. Patterns also have a long tradition in computer science. Gamma et al. [24] applied the same concepts to software engineering and described 23 design patterns. In the workflow area, patterns have been introduced for analyzing the expressiveness of process meta models [6, 25]. In this context, control flow patterns describe different constructs to specify activities and their ordering. In addition, workflow data patterns [7] provide ways for modeling the data aspect in PAISs, workflow resource patterns [8] describe how resources can be represented in workflows. Furthermore, patterns for describing typical control-flow changes [9] and service interactions were introduced [26]. Finally, activity patters cover the semantics of widespread business functions and business interactions respectively [27]. The introduction of workflow patterns has had significant impact on PAIS design and on the evaluation of PAISs and process languages. To evaluate the powerfulness of a PAIS regarding its ability to cope with time aspects, the existing workflow patterns are important, but not sufficient. In addition, a set of patterns addressing different time constraints are needed.

Most academic approaches on time support for PAISs focus on time features like verification of time constraints [2, 5, 1, 3, 28], escalation management [12] and scheduling support [13, 14]. The effect of ad-hoc changes on temporal constraints is investigated in [29]. A systematic investigation of requirements for time support from different heterogeneous application domains is missing so far.

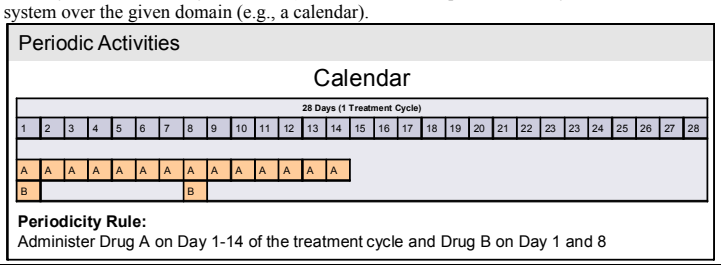
<table>
<tr><th colspan="2">TP10: Periodicity</th></tr>
<tr><td>Description</td><td>A particular activity, activity set, or process shall be performed periodically; i.e., according to a particular periodicity rule. Periodic implies some regularity, but does not necessarily mean equally distanced. A periodicity rule describes the reoccurrence pattern of the respective element (e.g., every Monday at 11:30) as well as start and end points (e.g., starting from next Monday, until end of the year, 5 times). The reoccurrence patterns usually use a reference system over the given domain (e.g., a calendar).
 
</td></tr>
<tr><td>Synonyms</td><td>Often referred to as “Recurrence” or “Appointment Series”</td></tr>
<tr><td rowspan="3">Design Choices</td><td>W.) Can the periodicity rule contain
a.) Only one date (e.g., Monday at 10:00) b.)
b.) More than 1 date (e.g., Monday morning and evening)</td></tr>
<tr><td>X.) Number of cycles
a.) Fixed / dynamic number of iterations b.) Depends on time lag and end date
c.) Depends on exit condition</td></tr>
<tr><td>Y.) May exceptions to the periodicity rule be specified (e.g., for leap years, public holidays)?
a.) Yes b.) No</td></tr>
<tr><td>Example</td><td>• Starting with next Monday group meetings will take place every two weeks at 11:30 (Design Choices W[a] X[c]).
• Each day at 7:00 the responsible assistant physician of the Gynaecological Clinic is informing the assistant medical director about the patients (Design Choices W[a] X[c]).
• Course ``Business Processes and Workflows" takes place every Monday from 8:00 to 11:00 starting on Oct 6th and ending on Jan 26th. On Dec 8th, 22nd, 29th and on Jan 5th there will be no lectures taking place (Design Choices W[a] X[b] Y[a]).
• Stationary chemotherapy usually comprises 6 treatments which are performed every 14 days. At the end of one treatment cycle the date for the next chemotherapy is scheduled (Design Choices W[a] X[a]).
• Administer Drug A on day 1 to 14 of each of the 6 treatment cycles and Drug B on the 1st and the 8th day. At the end of each treatment cycle the starting date for the next cycle is scheduled (Design Choices W[a] X[a]).</td></tr>
<tr><td>Related Patterns</td><td>TP9 – Cyclic Elements</td></tr>
</table>

Fig. 12. TP10 - Periodicity

7 Summary and Outlook

This paper has proposed 10 time patterns to foster the selection of appropriate PAIS-enabling technologies and to facilitate the comparison of process management systems, calendar systems and project planning tools regarding their ability to cope with time constraints. We have shown that the suggested time patterns are highly relevant in practice and complement existing workflow patterns with another fundamental dimension that is particularly important in the context of long-running business processes. Furthermore, our evaluation has proven that the support of time constraints has been neglected in commercial process management systems so far, whereas academic prototypes, calendar systems and project planning tools provide more sophisticated time support. From the evaluation of the latter system categories, we can further learn that a closer inte-

Patterns	Calendar Systems	Project Management	Standards	Commercial		Academic		
	Outlook 2007, Lightning 0.9, Google Calendar	MS Project	BPMN	MQ Workflow	TIBCO iProcess Modeller	Eder et al.	Bettini et al.	Combi et al.
General Design Choices	A[c], B[a, b]	A[a, c], B[a, b]	A[a, c], B[a?, b?, c?]	A[a, c], B[a]	A[a, c], B[a, b]	A[a, b, c]	A[a, b?, c?], B[a, c]	A[a, b?, c?], B[a, c*]
Category I: Durations and Time Lags								
TP1 – Time Lags between Events	–	–	–	–	–	–	C[a, b, c], D[a]	D[a], C[a*, b*, c*]
TP2 – Durations	E[a], F[b]	E[a], F[b, d][1]	E[a, c*], F[b]	E[a], F[b]	E[a], F[b]	E[a, c*], F[a, b, d][2]	E[a, c], F[a, b, c]	E[a, c], F[a, b, c]
TP3 – Time Lags between Activities	–	G[a, b, c, d], H[a, d][3], I[a], J[a]	G[c], H[a, b*], I[a], J[a]	–	–	G[a*,b*,c*,d], H[a, b, c], I[a], J[b]	G[a, b, c, d], H[a, b, c], I[a], J[b]	G[a, b, c, d], H[a, b, c], I[a], J[b]
Category II: Restrictions of Process Execution Points								
TP4 – Fixed Date Elements	K[a, b*], L[a, b, c]	K[a, c], L[a]	K[a, b, c], L[c]	–	–	K[a*], L[b]	K[a], L[a, b, c]	K[a, b], L[c, d]
TP5 – Schedule Restricted Elements	–	M[a], N[b], O[a]	–	–	–	M[a], N[a, b], O[a]	–	M[a], N[a, b]
TP6 – Time Based Restrictions	–	–	–	–	–	–	–	–
TP7 – Validity Period	–	–	–	R[b], S[c]	–	–	–	–
Category III: Variability								
TP8 – Time Dependent Variability	–	–	T[a, b]	–	–	?	?	?
Category IV: Reoccurring Process Elements								
TP9 – Cyclic Elements	U[a], V[a, b]	U[a], V[a, b]	U[a*, c*], V[a, c]	–	–	–	–	U[a, b], V[a, b, c]
TP10 – Periodicity	W[a], X[a, b], Y[a*]	W[a], X[a, b], Y[a*]	W[a*, b*], X[a, c], Y[a*]	–	–	–	–	W[a], X[a, b, c]

Symbols:
A[x] Support of Design Choice A with option x.
A[x*] Option x is partially supported
– Not supported
? Not specified

[1] Only one duration value per activity is supported, which can either be maximum or average
[2] Only one duration value is support. It may be minimum, maximum or average, depending on the concrete implementation
[3] Only one Time Lag value is supported which can either be minimum or average

Fig. 13. Evaluation Results

gration of the different technologies offers promising perspectives in respect to full coverage of the identified time patterns. In our future work we will formalize the time patterns and provide a reference implementation. Furthermore, we will conduct a comprehensive study of time support features (e.g., verification of time constraints, escalation management, scheduling support), in addition to the proposed time patterns, and also consider the resource dimension in this context.

References

1. Combi, C., Gozzi, M., Juarez, J., Oliboni, B., Pozzi, G.: Conceptual modeling of temporal clinicalworkflows. In: Proc. TIME'07. (2007) 70 – 81
2. Marjanovic, O., Orlowska, M.E.: On modeling and verification of temporal constraints in production workflows. Knowl. Inf. Syst. **1** (1999) 157–192
3. Eder, J., Panagos, E., Rabinovich, M.: Time constraints in workflow systems. In: CAiSE'99. (1999) 286–300
4. Dadam, P., Reichert, M., Kuhn, K.: Clinical workflows – the killer application for process-oriented information systems? In: Proc. Int'l Conf. on Business Information Systems (BIS'00), Poznan, Poland (2000) 36–59
5. Bettini, C., Wang, X.S., Jajodia, S.: Temporal reasoning in workflow systems. In: Distributed and Parallel Databases. (2002) 269–306
6. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow Patterns. Distributed and Parallel Databases **14** (2003) 5–51
7. Russell, N., ter Hofstede, A., Edmond, D., van der Aalst, W.: Workflow Data Patterns. Technical Report FIT-TR-2004-01, Queensland Univ. of Techn. (2004)
8. Russell, N., ter Hofstede, A., Edmond, D., van der Aalst, W.: Workflow Resource Patterns. Technical Report WP 127, Eindhoven Univ. of Technology (2004)
9. Weber, B., Reichert, M., Rinderle-Ma, S.: Change Patterns and Change Support Features -Enhancing Flexibility in Process-Aware Information Systems. Data and Knoweldge Engineering **66** (2008) 438–466
10. Russell, N., van der Aalst, W., ter Hofstede, A.: Exception Handling Patterns in Process-Aware Information Systems. In: Proc. CAiSE'06. (2006) 288–302
11. Rinderle-Ma, S., Reichert, M., Weber, B.: On the formal semantics of change patterns in process-aware information systems. In: Proc. 27th Int'l Conference on Conceptual Modeling (ER'08). (2008) 279–293
12. van der Aalst, W.M.P., Rosemann, M., Dumas, M.: Deadline-based escalation in process-aware information systems. Decision Support Systems **43** (2007) 492–511
13. Combi, C., Pozzi, G.: Task scheduling for a temporalworkflow management system. In: Proc. TIME'06. (2006) 61–68
14. Eder, J., Pichler, H., Gruber, W., Ninaus, M.: Personal schedules for workflow systems. In: Proc. BPM'03. (2003) 216–231
15. Schultheiß, B., Meyer, J., Mangold, R., Zemmler, T., Reichert, M.: Prozessentwurf für den Ablauf einer stationären Chemotherapie (in German). (1996)
16. Schultheiß, B., Meyer, J., Mangold, R., Zemmler, T., Reichert, M.: Prozessentwurf am Beispiel eines Ablaufs aus dem OP. (in German). (1996)
17. Schultheiß, B., Meyer, J., Mangold, R., Zemmler, T., Reichert, M.: Prozessentwurf für den Ablauf einer ambulante Chemotherapie (in German). (1996)

18. Konyen, I., Reichert, M., Schultheiß, B., Frank, S., Mangold, R.: Prozessentwurf für den Bereich der minimal invasiven Chirurgie (in German). (1996)
19. Schultheiß, B., Frank, S., Reichert, M.: Prototypische Implementierung des Prozessentwurfs für den Bereich der minimal invasiven Chirurgie mit WorkParty (in German). (1996)
20. German Association of the Automotive Industry (VDA): Engineering Change Management. Part 1: Engineering Change Request (ECR), V 1.1., Doc. No. 4965, Dec 2005 (2005)
21. Bobrik, R.: Konfigurierbare Visualisierung komplexer Prozessmodelle. PhD thesis, Univ. of Ulm (2008)
22. Allen, J.F.: Maintaining knowledge about temporal intervals. Communications of the ACM (1983) 832–843
23. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language. Oxford University Press, New York (1977)
24. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley (1995)
25. Puhlmann, F., Weske, M.: Using the Pi-Calculus for Formalizing Workflow Patterns. In: Proc. BPM'05. (2005) 153–168
26. Barros, A., Dumas, M., ter Hofstede, A.: Service Interaction Patterns. In: Proc. BPM'05. (2005) 302–318
27. Thom, L., Reichert, M., Chiao, C.M., Iochpe, C., Hess, G.N.: Inventing less, reusing more and adding intelligence to business process modeling. In: 19th International Conference on Database and Expert Systems Applications (DEXA '08). (2008)
28. Kafeza, E., Karlapalem, K.: Gaining control over time in workflow management applications. In: DEXA'00. (2000) 232–241
29. Sadiq, W., Marjanovic, O., Orlowska, M.E.: Managing change and time in dynamic workflow processes. Int. J. Cooperative Inf. Syst. **9** (2000) 93–116
30. Object Management Group: Business Process Modeling Notation (BPMN) Version 1.2. http://www.omg.org/spec/BPMN/1.2 (2009)

A Evaluation Details

A.1 Evaluation Results: Outlook - Lightening - Google Calendar

MS Outlook 2007, *Mozilla Lightening 0.9* and *Google Calendar* are widely used calendar systems. Due to their similarities in respect to time support the evaluation results of these three calendar systems are discussed in conjunction with each other.

General Design Choices. In terms of general design choices the evaluated calendar systems support Design Choice A with Options a and c, while Option b is not applicable. Design Choice B, in turn, is supported with Options a and b (Design Choice B[a,b]). As basic granularities, Day, Week, Month and Year are supported. In addition, Minute and Hour are supported when specifying durations. All calendar systems support system-defined granularities. *Google Calendar* supports Working Days, each Monday, Wednesday and Friday as well as each Tuesday and Thursday. *Lightening*, in turn, supports Bi-Weekly and

Working Days while *Outlook* only supports Working Days. For Design Choice B only Option c is supported (Design Choice B[c]).
Category I - Durations and Time Lags. The evaluated calendar systems only support Pattern TP2 (*Durations*) from Category I. Thereby, durations can only be specified for single activities (i.e., tasks) or events (i.e., calendar items) (Design Choice E[a]) and are treated as maximum values (Design Choice F[b]).
Category II - Restrictions of Process Execution Points. In this pattern category support is provided for Pattern TP4 (*Fixed Date Elements*). In particular, the pattern can be applied to single activities. *Google Calendar* and *Outlook* additionally support the invitation of different users to appointments. This can be considered as partial support for multiple instances of a particular activity (Design Choice K[a,b*[1]]). Both *Outlook* and *Lightening* provide full support for Design Choice L, as start dates, latest completion dates and earliest start dates can be specified (Design Choice L[a,b,c]). *Google Calendar*, in turn, only provides support for Options a and c (Design Choice L[a,c])
Category III - Variability. This pattern category is not supported by any of the calendar systems.
Category IV - Reoccurring Process Elements. This pattern category is supported pretty well by all evaluated calendar systems. Pattern TP9 (*Cyclic Elements*) is supported with Design Choice U[a], i.e., time lags between the different cycles are always the same. The number of cycles can either be fixed (i.e., 5 cyles) or be calculated depending on the time lag and the end date (e.g., every 2 weeks until the end of the year) (Design Choice V[a,b]). Pattern TP10 (*Periodicity*), in turn, is supported with Design Choice W[a], i.e., the periodicity rule can only contain one date. Like for TP9 the number of cycles can either be fixed (i.e., 5 cyles) or be calculated depending on the time lag between cycles and the end date (e.g., every 2 weeks until the end of the year) (Design Choice X[a,b]). Finally, Design Choice Y[a*] is only partially supported as exceptions to a periodicity rule have to be defined manually by deleting single appointments.

A.2 Evaluation Results: MS Project 2007

MS Project is a commonly used commercial project planning system.

General Design Choices. In terms of general design choices *MS Project* supports Design Choice A with Options a and c, while Option b is not applicable (Design Choice A[a,c]). Design Choice B, in turn, is supported with Options a and b (Design Choice B[a,b]). As basic granularities Day, Week, Month, and Year are supported. In addition, Minute and Hour are supported when specifying durations. In terms of system-defined granularities *MS Project* supports Working Days as well.
Category I - Durations and Time Lags. *MS Project* supports Pattern TP2 (*Durations*) and Pattern TP3 (*Time Lags between Activities*) from Category I.

[1] A * means, the respective design choice is only partially supported based on some workaround.

In the context of Pattern TP2 durations can be only specified for single activities (i.e., tasks) (Design Choice E[a]), and can either be average values or maximum ones (Design Choice F[b,d]). Note that only one duration value per activity is supported (i.e., it is not possible to assign both a maximum and an average duration to the same activity). Regarding Pattern TP3, Design Choice G is fully supported; i.e., start-start relations, start-end relations, end-start relations, and end-end relations can be expressed (Design Choice G[a,b,c,d]). Time lags can either be minimum values or average values (Design Choice F[a,d]) and are quantitatively expressed (Design Choice I[a]). Finally, time lags can only be expressed between two directly succeeding activities (Design Choice J[a]), since the control flow is modelled through time dependencies.

Category II - Restrictions of Process Execution Points. In this pattern category, support is provided for Patterns TP4 (*Fixed Date Elements*) and TP5 (*Schedule Restricted Elements*). Pattern TP4 can be applied to single activities, but also to a whole process instance (i.e., an entire project) (Design Choice K[a,c]). Design Choice L is supported with Option a, allowing for the specification of execution dates only (Design Choice L[a]). Regarding Pattern TP5, in turn, Design Choice M is supported for single activities (i.e., for each activity a special calendar can be created) (Design Choice M[a]). Schedule entries correspond to time frames (i.e., special working hours can be specified) (Design Choice N[b]). Finally, it is possible to specify exceptions like days off work (Design Choice O[a]).

Category III - Variability. This pattern category is not supported by *MS Project*.

Category IV - Reoccurring Process Elements. This pattern category is resonably supported. For Pattern TP9 (*Cyclic Elements*), Design Choice U is supported with Option a, i.e., time lags between different cycles are always the same (Design Choice U[a]). In addition, support for Design Choice V[a,b] is provided – the number of cycles can either be fixed (i.e., 5 cyles) or be calculated depending on the time lag and the end date (e.g., every Monday at 11:30 until the end of the year). Pattern TP10 (*Periodicity*), in turn, is supported with Design Choice W[a], i.e., any periodicity rule may only contain one date. Like for TP9 the number of cycles can either be fixed (i.e., 5 cyles) or be calculated depending on the time lag and the end date (Design Choice X[a,b]). Finally, Design Choice Y[a] is only partially supported as exceptions to a periodicity rule have to be manually defined by deleting single appointments.

A.3 Evaluation Results: Business Process Modelling Notation 1.2

The *Business Process Modelling Notation (BPMN)* [30] is a process modelling standard published by the Object Management Group (OMG). It was specifically designed to provide standardized notations and diagramming conventions for the description of business processes.

General Design Choices. Using *BPMN* it becomes possible to set time parameters during build-time as well as determining them during run-time at the

time they are needed for further process execution (Design Choice A[a,c]). Concerning Design Choice B, it might be possible to use basic time granularities, as well as system-defined and user-defined ones (Design Choice B[a?[2],b?,c?]); since the formalism used for specifying time parameters is not preassigned no definite conclusion can be made.

Category I - Durations and Time Lags. *BPMN* supports time pattern TP2 with Design Choices E[a,c*] and F[b], i.e., it allows to specify the maximum duration of an activity by attaching an Intermediate-Timer-Event to the activity. By adding a cancelling discriminator [6] together with an Intermediate-Timer-Event in parallel to the whole process it also becomes possible to emulate Option c of Design Choice E. Regarding Pattern TP3, *BPMN* only allows to express end-start relations (Design Choice G[c]). Regarding Design Choice H, Option a is fully supported by adding an Intermediate-Timer-Event on the sequence flow connecting the activities; Option b may be emulated by adding an event-based decision between the two activities with an Intermediate-Timer-Event parallel to the second activity. Time Lags are quantitatively expressed (Design Choice I[a]) and only relations between two directly succeeding activities are supported (Design Choice J[a]).

Category II - Restrictions of Process Execution Points. In this pattern category *BPMN* only provides support for pattern TP4 (*Fixed Date Elements*). Design Choice K is fully supported, i.e., it is possible to add a fixed date to an activity or multi-instance activity (by putting an Intermediate-Timer-Event on the sequence flow leading to the respective element) or to a process (by using a Start-Timer-Event) (Design Choice K[a,b,c]). Due to the modelling technique using timers, for Design Choice L only Option c is supported.

Category III - Variability. This Pattern Category is supported by using an event-based XOR in combination with an Intermediate-Timer-Event (Design Choice T[a,b]).

Category IV - Reoccurring Process Elements. *BPMN* supports this pattern category to a certain degree. Concerning TP9 (*Cyclic Elements*) design choice U[a*,c*] is partially supported by adding an Intermediate-Timer-Event on the sequence flow connecting the iterations. Additionally, support for Design Choice V[a,c] is provided as the number of cycles can either be fixed (e.g. 5 iterations) or be dependent on an exit condition. Regarding pattern TP10 (*Periodicity*), Design Choice W is partially supported with Option a, whereas support for Option b depends on the formalism used for specifiying time, which as stated before, is not preassigned. As with TP9 the number of cycles may either be fixed or be dependent on an exit condition (Design Choice X[a,c]). Whether or not exceptions to the periodicity rule are supported again depends on the formalism used for specifying time constructs.

[2] A ? means, that based on the available material we could not decide whether or not the respective design choice is supported.

A.4 Evaluation Results: IBM WebShere MQ Workflow Buildtime 3.4

MQ Workflow Buildtime is the graphical process definition tool that ships with IBM WebSphere MQ Workflow.

General Design Choices. Regarding general design choices, *MQ Workflow* supports setting the time parameters during build-time or specifying a data container which provides them during runtime (Design Choice A[a,c]). Furthermore, *MQ Workflow* supports the basic time granularities Year, Month, Week, Day, Hour, Minute and Second (Design Choice B[a]).
Category I - Durations and Time Lags. In Category I, *MQ Workflow* only supports pattern TP2. More precisely, it is possible to specify the maximum duration (Design Choice F[b]) of a single activity or process (Design Choice E[a,c]).
Category II - Restrictions of Process Execution Points. As the only one of the systems evaluated by us, *MQ Workflow* allows for the specification of a Validity Date (TP7). Thereby, it is possible to specify the earliest possible start date (Design Choice S[c]) of a process (Design Choice R[b]). None of the other patterns in this category is supported.
Category III - Variability. This pattern category is not supported by *MQ Workflow*.
Category IV - Reoccurring Process Elements. This pattern category is not supported by *MQ Workflow*.

A.5 Evaluation Results: Tibco iProcess Modeller 10.3.5

Tibco iProcess Modeller is one of the most popular workflow tools in practice.

General Design Choices. *Tibco iProcess Modeller* allows specifying time parameters during build-time as well as determining them during process execution (Design Choice A[a,c]). Concerning Design Choice B, *Tibco iProcess Modeller* supports basic time granularities; i.e., Year, Month, Week, Day, Hour, and Minute. In addition, it is possible to use Working Days when specifying time expressions (Design Choice B[a,b]).
Category I - Durations and Time Lags. From this category only Pattern TP2 (*Durations*) is supported. Thereby, it is only possible to specify the maximum (Design Choice F[b]) duration of a single activity (Design Choice E[a]).
Category II - Restrictions of Process Execution Points. *Tibco iProcess Modeller* does not allow for the restriction of process execution points in any way.
Category III - Variability. It is not possible to vary the control flow depending on time time aspects.
Category IV - Reoccurring Process Elements. None of the patterns of Category IV is supported.

A.6 Evaluation Results: Eder et al. [3]

Eder et al. [3] discusses an approach for calculating activity deadlines such that all time constraints are satisfied and the overall process deadline can be met.

General Design Choices. Concerning general design choices, time parameters may be set during build-time, be fixed at process instantiation time or be determined during runtime (Design Choice A[a,b,c]). Only one basic granularity is considered, i.e., [3] does not provide direct support for Design Choice B.
Category I - Durations and Time Lags. In this category, Time Patterns TP2 (*Durations*) and TP3 (*Time Lags between Activities*) are considered. Regarding Design Choice E, Option a is supported; Option c can be simulated by using a time lag between the first and the last activity of the process (Design Choice E[a, c*]). Regarding Design Choice F, all three Options are supported (Design Choice F[a,b,c]), but only one duration value may be specified per activity, the concrete kind of which is set by the respective implementation. Thus each implementation only supports one of these three Options. Time Lags between activities (TP3) may be specified in terms of end-end relations. However, since the durations of the activities are considered to be deterministic, it is possible to also simulate start-start, start-end and end-start relations by adding the duration of the first and/or substracting the duration of the second activity from the time lag (Design Choice G[a*,b*,c*,d]). Time Lags can be represented as minimum, as maximum or as time intervall (Design Choice H[a,b,c]), and are expressed quantitatively (Design Choice I[a]). Finally, time lags can be specified between any two activities (Design Choice J[b]).
Category II - Restrictions of Process Execution Points. [3] indicates support of Fixed Date Elements (TP4) by using a schedule with only one valid date. However, the case in which for a specific point in time no further date is available from the Schedule is not considered in the algorithm (Design Choice K[a*]). Since [3] just considers end events of activities it is possible to specify the latest completion time of an activity (Design Choice L[b]). As aforementioned, TP5 (*Schedule Restricted Elements*) is supported. In particular it is possible to specify a schedule for an activity (Design Choice M[a]), which can either consist of several discrete points in time or time frames (Design Choice N[a,b]). Additionaly these schedules can support exceptions (Design Choice O[a]). Detailed information on the implementation of the schedules is not available.
Category III - Variability. Support for this pattern is not explicitly considered, but depends on the underlying workflow management system.
Category IV - Reoccurring Process Elements. Since [3] does not consider the repetitive execution of process elements, no support for this category is given.

A.7 Evaluation Results: Bettini et al. [5]

Bettini et al. [5] investigate the calculation of enactment schedules for activities, which guarantee, that all temporal dependencies are met.

General Design Choices. Regarding general design choices, [5] supports Design Choice A with Option a, while Options b and c are not discussed (Design Choice A[a,b?,c?]). Moreover support for basic time granularities as well as user-defined granularities (e.g. business days) is provided (Design Choice B[a,c]).
Category I - Durations and Time Lags. Pattern Category IV is broadly supported by this approach. Pattern TP1 is supported with Design Choice C[a,b,c], i.e., time lags between events can be represented as minimum, as maximum or as time intervall. Design Choice D is supported with Option a, i.e., time lags are expressed quantitativly. *Durations* (TP2) can be specified for a single activity or for a whole process (by specifying a time lag between the start of the first and the end of the last activity) (Design Choice E[a,c]). Like for TP1, minimum, maximum or time intervalls can be set for durations (Design Choice F[a,b,c]). For Pattern TP3 (*Time Lags between Activities*) all four types of relations are supported, i.e., start-start, start-end, end-start, and end-relation (Design Choice G[a,b,c,d]). Again, time lags can be set in terms of minimum, maximum or time intervall (Design Choice H[a,b,c]). They are specified in a quantitative way (Design Choice I[a]), and can be expressed between any two activities (Design Choice J[b]).
Category II - Restrictions of Process Execution Points. [5] supports the specification of *Fixed Date Elements* (TP4) for single activity instances (Design Choice K[a]) by adding a time lag between an artifical event at time point 01.01.0000 and the activity in question. Thereby it is possible to specify the execution date, the latest completion date, and the earliest start date (Design Choice L[a,b,c]) of the activity.
Category III - Variability. Support for this pattern category is not explicitly expressed, but depends on the used workflow management system.
Category IV - Reoccurring Process Elements. [5] does not consider the repetitive execution of process elements, i.e., no support for this pattern category is provided.

A.8 Evaluation Results: Combi et al. [1]

Combi et al. [1] discusses the conceptual modelling of temporal constraints in the medical domain and provides rather broad support for our time patterns.

General Design Choices. In terms of general design choices, [1] supports Design Choice A with Option a, while Option b and Option c are not discussed (Design Choice A[a,b?,c?]). Combi et al. [1] considers different time granularities, slicing the time domain into a sequence of granules. However, no details are provided on which granules are supported. User-defined granularities are partially supported (i.e., only granularities without laps are considered) (Design Choice B[a,c*]).
Category I - Durations and Time Lags. [1] partially supports Pattern TP1 (*Time Lags between Events*), supports Pattern TP2 (*Durations*) and Pattern TP3 (*Time Lags between Activities*). Pattern TP1 is supported with Design Choice C[a*,b*,c*]. Time lags cannot be defined between arbitrary events as at

least on of the events has to be a start or completion event of an activity (e.g., a time lag between the reaching of a milestone and the start of a subsequent activity). Time lags are specified in a quantitative way (Design Choice D[a]). Durations in the context of Pattern TP2 can be specified for single activities as well as for an entire process (by specifying a time lag between the start of the first and the end of the last activity) (Design Choice E[a,c]). Durations can either be minimum values, maximum values, or time intervals (Design Choice F[a,b,c]). For Pattern TP3, Design Choice G is fully supported, i.e., start-start relations, start-end relations, end-start relations, and end-end relations can be expressed (Design Choice G[a,b,c,d]). Time lags can either be minimum values, maximum values or time intervals (Design Choice H[a,b,c]), and are quantitatively expressed (Design Choice I[a]). Finally, time lags can be expressed between any two activities (Design Choice J[b]).

Category II - Restrictions of Process Execution Points. In this pattern category support is provided for Pattern TP4 (*Fixed Date Elements*) and Pattern TP5 (*Schedule Restricted Elements*). Pattern TP4 can be applied to single activities, but also to multiple instances of a single activity (i.e., Multitask with Fixed Date Element) (Design Choice K[a,c]). Design Choice L is supported with Options b and c, allowing for the specification of execution dates only (Design Choice L[b,c]). Thereby, absolute constraints restrict the intervall during which an activity can be performed. For Pattern TP5, in turn, Design Choice M is supported for single activities (i.e., for each activity a special calendar can be created) (Design Choice M[a]). Schedule entries can be discrete points as well as time frames (Design Choice N[a,b]). Support for exceptions, in turn, is not provided (Design Choice O[b]).

Category III - Variability. Support for this pattern is not explicitly addressed, but depends on the underlying workflow management system.

Category IV - Reoccurring Process Elements. This pattern category is supported very well by this approach. For Pattern TP9 (*Cyclic Elements*), Design Choice U is supported with Option a; i.e., the time lags between the different cycles are always the same; Option b is supported as well, i.e., time lags can be fuzzy (Design Choice U[a,b]). In addition, support for Design Choice V[a,b,c] is provided as the number of cycles can either be fixed (i.e., 5 cyles), be calculated depending on the time lag and the end date, or depend on the exit condition. Pattern TP10 (*Periodicity*), in turn, is supported with Design Choice W[a], i.e., periodic constraints on loop activities can be expressed. Like for TP9 the number of cycles can either be fixed (i.e., 5 cyles), be calculated depending on the time lag and the end date, or depend on the exit condition (Design Choice X[a,b,c]). Finally, it is not possible to specify exceptions to a periodicity rule (Design Choice Y[b]).

Liste der bisher erschienenen Ulmer Informatik-Berichte

Einige davon sind per FTP von `ftp.informatik.uni-ulm.de` erhältlich
Die mit * markierten Berichte sind vergriffen

List of technical reports published by the University of Ulm

Some of them are available by FTP from `ftp.informatik.uni-ulm.de`
Reports marked with * are out of print

91-01 *Ker-I Ko, P. Orponen, U. Schöning, O. Watanabe*
Instance Complexity

91-02* *K. Gladitz, H. Fassbender, H. Vogler*
Compiler-Based Implementation of Syntax-Directed Functional Programming

91-03* *Alfons Geser*
Relative Termination

91-04* *J. Köbler, U. Schöning, J. Toran*
Graph Isomorphism is low for PP

91-05 *Johannes Köbler, Thomas Thierauf*
Complexity Restricted Advice Functions

91-06* *Uwe Schöning*
Recent Highlights in Structural Complexity Theory

91-07* *F. Green, J. Köbler, J. Toran*
The Power of Middle Bit

91-08* *V.Arvind, Y. Han, L. Hamachandra, J. Köbler, A. Lozano, M. Mundhenk, A. Ogiwara, U. Schöning, R. Silvestri, T. Thierauf*
Reductions for Sets of Low Information Content

92-01* *Vikraman Arvind, Johannes Köbler, Martin Mundhenk*
On Bounded Truth-Table and Conjunctive Reductions to Sparse and Tally Sets

92-02* *Thomas Noll, Heiko Vogler*
Top-down Parsing with Simulataneous Evaluation of Noncircular Attribute Grammars

92-03 *Fakultät für Informatik*
17. Workshop über Komplexitätstheorie, effiziente Algorithmen und Datenstrukturen

92-04* *V. Arvind, J. Köbler, M. Mundhenk*
Lowness and the Complexity of Sparse and Tally Descriptions

92-05* *Johannes Köbler*
Locating P/poly Optimally in the Extended Low Hierarchy

92-06* *Armin Kühnemann, Heiko Vogler*
Synthesized and inherited functions -a new computational model for syntax-directed semantics

92-07* *Heinz Fassbender, Heiko Vogler*
A Universal Unification Algorithm Based on Unification-Driven Leftmost Outermost Narrowing

92-08* *Uwe Schöning*
On Random Reductions from Sparse Sets to Tally Sets

92-09* *Hermann von Hasseln, Laura Martignon*
Consistency in Stochastic Network

92-10 *Michael Schmitt*
A Slightly Improved Upper Bound on the Size of Weights Sufficient to Represent Any Linearly Separable Boolean Function

92-11 *Johannes Köbler, Seinosuke Toda*
On the Power of Generalized MOD-Classes

92-12 *V. Arvind, J. Köbler, M. Mundhenk*
Reliable Reductions, High Sets and Low Sets

92-13 *Alfons Geser*
On a monotonic semantic path ordering

92-14* *Joost Engelfriet, Heiko Vogler*
The Translation Power of Top-Down Tree-To-Graph Transducers

93-01 *Alfred Lupper, Konrad Froitzheim*
AppleTalk Link Access Protocol basierend auf dem Abstract Personal Communications Manager

93-02 *M.H. Scholl, C. Laasch, C. Rich, H.-J. Schek, M. Tresch*
The COCOON Object Model

93-03 *Thomas Thierauf, Seinosuke Toda, Osamu Watanabe*
On Sets Bounded Truth-Table Reducible to P-selective Sets

93-04 *Jin-Yi Cai, Frederic Green, Thomas Thierauf*
On the Correlation of Symmetric Functions

93-05 *K.Kuhn, M.Reichert, M. Nathe, T. Beuter, C. Heinlein, P. Dadam*
A Conceptual Approach to an Open Hospital Information System

93-06 *Klaus Gaßner*
Rechnerunterstützung für die konzeptuelle Modellierung

93-07 *Ullrich Keßler, Peter Dadam*
Towards Customizable, Flexible Storage Structures for Complex Objects

94-01 *Michael Schmitt*
On the Complexity of Consistency Problems for Neurons with Binary Weights

94-02 *Armin Kühnemann, Heiko Vogler*
A Pumping Lemma for Output Languages of Attributed Tree Transducers

94-03 *Harry Buhrman, Jim Kadin, Thomas Thierauf*
On Functions Computable with Nonadaptive Queries to NP

94-04 *Heinz Faßbender, Heiko Vogler, Andrea Wedel*
Implementation of a Deterministic Partial E-Unification Algorithm for Macro Tree Transducers

94-05 *V. Arvind, J. Köbler, R. Schuler*
On Helping and Interactive Proof Systems

94-06 *Christian Kalus, Peter Dadam*
Incorporating record subtyping into a relational data model

94-07 *Markus Tresch, Marc H. Scholl*
A Classification of Multi-Database Languages

94-08 *Friedrich von Henke, Harald Rueß*
Arbeitstreffen Typtheorie: Zusammenfassung der Beiträge

94-09 *F.W. von Henke, A. Dold, H. Rueß, D. Schwier, M. Strecker*
Construction and Deduction Methods for the Formal Development of Software

94-10 *Axel Dold*
Formalisierung schematischer Algorithmen

94-11 *Johannes Köbler, Osamu Watanabe*
New Collapse Consequences of NP Having Small Circuits

94-12 *Rainer Schuler*
On Average Polynomial Time

94-13 *Rainer Schuler, Osamu Watanabe*
Towards Average-Case Complexity Analysis of NP Optimization Problems

94-14 *Wolfram Schulte, Ton Vullinghs*
Linking Reactive Software to the X-Window System

94-15 *Alfred Lupper*
Namensverwaltung und Adressierung in Distributed Shared Memory-Systemen

94-16 *Robert Regn*
Verteilte Unix-Betriebssysteme

94-17 *Helmuth Partsch*
Again on Recognition and Parsing of Context-Free Grammars:
Two Exercises in Transformational Programming

94-18 *Helmuth Partsch*
Transformational Development of Data-Parallel Algorithms: an Example

95-01 *Oleg Verbitsky*
On the Largest Common Subgraph Problem

95-02 *Uwe Schöning*
Complexity of Presburger Arithmetic with Fixed Quantifier Dimension

95-03 *Harry Buhrman,Thomas Thierauf*
The Complexity of Generating and Checking Proofs of Membership

95-04 *Rainer Schuler, Tomoyuki Yamakami*
Structural Average Case Complexity

95-05 *Klaus Achatz, Wolfram Schulte*
Architecture Indepentent Massive Parallelization of Divide-And-Conquer Algorithms

95-06 *Christoph Karg, Rainer Schuler*
Structure in Average Case Complexity

95-07 *P. Dadam, K. Kuhn, M. Reichert, T. Beuter, M. Nathe*
ADEPT: Ein integrierender Ansatz zur Entwicklung flexibler, zuverlässiger kooperierender Assistenzsysteme in klinischen Anwendungsumgebungen

95-08 *Jürgen Kehrer, Peter Schulthess*
Aufbereitung von gescannten Röntgenbildern zur filmlosen Diagnostik

95-09 *Hans-Jörg Burtschick, Wolfgang Lindner*
On Sets Turing Reducible to P-Selective Sets

95-10 *Boris Hartmann*
Berücksichtigung lokaler Randbedingung bei globaler Zieloptimierung mit neuronalen Netzen am Beispiel Truck Backer-Upper

95-12 *Klaus Achatz, Wolfram Schulte*
Massive Parallelization of Divide-and-Conquer Algorithms over Powerlists

95-13 *Andrea Mößle, Heiko Vogler*
Efficient Call-by-value Evaluation Strategy of Primitive Recursive Program Schemes

95-14 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
A Generic Specification for Verifying Peephole Optimizations

96-01 *Ercüment Canver, Jan-Tecker Gayen, Adam Moik*
Formale Entwicklung der Steuerungssoftware für eine elektrisch ortsbediente Weiche mit VSE

96-02 *Bernhard Nebel*
Solving Hard Qualitative Temporal Reasoning Problems: Evaluating the Efficiency of Using the ORD-Horn Class

96-03 *Ton Vullinghs, Wolfram Schulte, Thilo Schwinn*
An Introduction to TkGofer

96-04 *Thomas Beuter, Peter Dadam*
Anwendungsspezifische Anforderungen an Workflow-Mangement-Systeme am Beispiel der Domäne Concurrent-Engineering

96-05 *Gerhard Schellhorn, Wolfgang Ahrendt*
Verification of a Prolog Compiler - First Steps with KIV

96-06 *Manindra Agrawal, Thomas Thierauf*
Satisfiability Problems

96-07 *Vikraman Arvind, Jacobo Torán*
A nonadaptive NC Checker for Permutation Group Intersection

96-08 *David Cyrluk, Oliver Möller, Harald Rueß*
An Efficient Decision Procedure for a Theory of Fix-Sized Bitvectors with Composition and Extraction

96-09 *Bernd Biechele, Dietmar Ernst, Frank Houdek, Joachim Schmid, Wolfram Schulte*
Erfahrungen bei der Modellierung eingebetteter Systeme mit verschiedenen SA/RT–Ansätzen

96-10 *Falk Bartels, Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Formalizing Fixed-Point Theory in PVS

96-11 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Mechanized Semantics of Simple Imperative Programming Constructs

96-12 *Axel Dold, Friedrich W. von Henke, Holger Pfeifer, Harald Rueß*
Generic Compilation Schemes for Simple Programming Constructs

96-13 *Klaus Achatz, Helmuth Partsch*
From Descriptive Specifications to Operational ones: A Powerful Transformation Rule, its Applications and Variants

97-01 *Jochen Messner*
Pattern Matching in Trace Monoids

97-02 *Wolfgang Lindner, Rainer Schuler*
A Small Span Theorem within P

97-03 *Thomas Bauer, Peter Dadam*
A Distributed Execution Environment for Large-Scale Workflow Management Systems with Subnets and Server Migration

97-04 *Christian Heinlein, Peter Dadam*
Interaction Expressions - A Powerful Formalism for Describing Inter-Workflow Dependencies

97-05 *Vikraman Arvind, Johannes Köbler*
On Pseudorandomness and Resource-Bounded Measure

97-06 *Gerhard Partsch*
Punkt-zu-Punkt- und Mehrpunkt-basierende LAN-Integrationsstrategien für den digitalen Mobilfunkstandard DECT

97-07 *Manfred Reichert, Peter Dadam*
$ADEPT_{flex}$ - Supporting Dynamic Changes of Workflows Without Loosing Control

97-08 *Hans Braxmeier, Dietmar Ernst, Andrea Mößle, Heiko Vogler*
The Project NoName - A functional programming language with its development environment

97-09 *Christian Heinlein*
Grundlagen von Interaktionsausdrücken

97-10 *Christian Heinlein*
Graphische Repräsentation von Interaktionsausdrücken

97-11 *Christian Heinlein*
Sprachtheoretische Semantik von Interaktionsausdrücken

97-12 *Gerhard Schellhorn, Wolfgang Reif*
Proving Properties of Finite Enumerations: A Problem Set for Automated Theorem Provers

97-13 *Dietmar Ernst, Frank Houdek, Wolfram Schulte, Thilo Schwinn*
Experimenteller Vergleich statischer und dynamischer Softwareprüfung für eingebettete Systeme

97-14 *Wolfgang Reif, Gerhard Schellhorn*
Theorem Proving in Large Theories

97-15 *Thomas Wennekers*
Asymptotik rekurrenter neuronaler Netze mit zufälligen Kopplungen

97-16 *Peter Dadam, Klaus Kuhn, Manfred Reichert*
Clinical Workflows - The Killer Application for Process-oriented Information Systems?

97-17 *Mohammad Ali Livani, Jörg Kaiser*
EDF Consensus on CAN Bus Access in Dynamic Real-Time Applications

97-18 *Johannes Köbler,Rainer Schuler*
Using Efficient Average-Case Algorithms to Collapse Worst-Case Complexity Classes

98-01 *Daniela Damm, Lutz Claes, Friedrich W. von Henke, Alexander Seitz, Adelinde Uhrmacher, Steffen Wolf*
Ein fallbasiertes System für die Interpretation von Literatur zur Knochenheilung

98-02 *Thomas Bauer, Peter Dadam*
Architekturen für skalierbare Workflow-Management-Systeme - Klassifikation und Analyse

98-03 *Marko Luther, Martin Strecker*
A guided tour through *Typelab*

98-04 *Heiko Neumann, Luiz Pessoa*
Visual Filling-in and Surface Property Reconstruction

98-05 *Ercüment Canver*
Formal Verification of a Coordinated Atomic Action Based Design

98-06 *Andreas Küchler*
On the Correspondence between Neural Folding Architectures and Tree Automata

98-07 *Heiko Neumann, Thorsten Hansen, Luiz Pessoa*
Interaction of ON and OFF Pathways for Visual Contrast Measurement

98-08 *Thomas Wennekers*
Synfire Graphs: From Spike Patterns to Automata of Spiking Neurons

98-09 *Thomas Bauer, Peter Dadam*
Variable Migration von Workflows in *ADEPT*

98-10 *Heiko Neumann, Wolfgang Sepp*
Recurrent V1 – V2 Interaction in Early Visual Boundary Processing

98-11 *Frank Houdek, Dietmar Ernst, Thilo Schwinn*
Prüfen von C–Code und Statmate/Matlab–Spezifikationen: Ein Experiment

98-12 *Gerhard Schellhorn*
Proving Properties of Directed Graphs: A Problem Set for Automated Theorem Provers

98-13 *Gerhard Schellhorn, Wolfgang Reif*
Theorems from Compiler Verification: A Problem Set for Automated Theorem Provers

98-14 *Mohammad Ali Livani*
SHARE: A Transparent Mechanism for Reliable Broadcast Delivery in CAN

98-15 *Mohammad Ali Livani, Jörg Kaiser*
Predictable Atomic Multicast in the Controller Area Network (CAN)

99-01 *Susanne Boll, Wolfgang Klas, Utz Westermann*
A Comparison of Multimedia Document Models Concerning Advanced Requirements

99-02 *Thomas Bauer, Peter Dadam*
Verteilungsmodelle für Workflow-Management-Systeme - Klassifikation und Simulation

99-03 *Uwe Schöning*
On the Complexity of Constraint Satisfaction

99-04 *Ercument Canver*
Model-Checking zur Analyse von Message Sequence Charts über Statecharts

99-05 *Johannes Köbler, Wolfgang Lindner, Rainer Schuler*
Derandomizing RP if Boolean Circuits are not Learnable

99-06 *Utz Westermann, Wolfgang Klas*
Architecture of a DataBlade Module for the Integrated Management of Multimedia Assets

99-07 *Peter Dadam, Manfred Reichert*
Enterprise-wide and Cross-enterprise Workflow Management: Concepts, Systems, Applications. Paderborn, Germany, October 6, 1999, GI–Workshop Proceedings, Informatik ’99

99-08 *Vikraman Arvind, Johannes Köbler*
Graph Isomorphism is Low for ZPP^{NP} and other Lowness results

99-09 *Thomas Bauer, Peter Dadam*
Efficient Distributed Workflow Management Based on Variable Server Assignments

2000-02 *Thomas Bauer, Peter Dadam*
Variable Serverzuordnungen und komplexe Bearbeiterzuordnungen im Workflow-Management-System ADEPT

2000-03 *Gregory Baratoff, Christian Toepfer, Heiko Neumann*
Combined space-variant maps for optical flow based navigation

2000-04 *Wolfgang Gehring*
Ein Rahmenwerk zur Einführung von Leistungspunktsystemen

2000-05 *Susanne Boll, Christian Heinlein, Wolfgang Klas, Jochen Wandel*
Intelligent Prefetching and Buffering for Interactive Streaming of MPEG Videos

2000-06 *Wolfgang Reif, Gerhard Schellhorn, Andreas Thums*
Fehlersuche in Formalen Spezifikationen

2000-07 *Gerhard Schellhorn, Wolfgang Reif (eds.)*
FM-Tools 2000: The 4th Workshop on Tools for System Design and Verification

2000-08 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Effiziente Durchführung von Prozessmigrationen in verteilten Workflow-Management-Systemen

2000-09 *Thomas Bauer, Peter Dadam*
Vermeidung von Überlastsituationen durch Replikation von Workflow-Servern in ADEPT

2000-10 *Thomas Bauer, Manfred Reichert, Peter Dadam*
Adaptives und verteiltes Workflow-Management

2000-11 *Christian Heinlein*
Workflow and Process Synchronization with Interaction Expressions and Graphs

2001-01 *Hubert Hug, Rainer Schuler*
DNA-based parallel computation of simple arithmetic

2001-02 *Friedhelm Schwenker, Hans A. Kestler, Günther Palm*
3-D Visual Object Classification with Hierarchical Radial Basis Function Networks

2001-03 *Hans A. Kestler, Friedhelm Schwenker, Günther Palm*
RBF network classification of ECGs as a potential marker for sudden cardiac death

2001-04 *Christian Dietrich, Friedhelm Schwenker, Klaus Riede, Günther Palm*
Classification of Bioacoustic Time Series Utilizing Pulse Detection, Time and Frequency Features and Data Fusion

2002-01 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Effiziente Verträglichkeitsprüfung und automatische Migration von Workflow-Instanzen bei der Evolution von Workflow-Schemata

2002-02 *Walter Guttmann*
Deriving an Applicative Heapsort Algorithm

2002-03 *Axel Dold, Friedrich W. von Henke, Vincent Vialard, Wolfgang Goerigk*
A Mechanically Verified Compiling Specification for a Realistic Compiler

2003-01 *Manfred Reichert, Stefanie Rinderle, Peter Dadam*
A Formal Framework for Workflow Type and Instance Changes Under Correctness Checks

2003-02 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
Supporting Workflow Schema Evolution By Efficient Compliance Checks

2003-03 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values

2003-04 *Stefanie Rinderle, Manfred Reichert, Peter Dadam*
On Dealing With Semantically Conflicting Business Process Changes.

2003-05 *Christian Heinlein*
Dynamic Class Methods in Java

2003-06 *Christian Heinlein*
Vertical, Horizontal, and Behavioural Extensibility of Software Systems

2003-07 *Christian Heinlein*
Safely Extending Procedure Types to Allow Nested Procedures as Values
(Corrected Version)

2003-08 *Changling Liu, Jörg Kaiser*
Survey of Mobile Ad Hoc Network Routing Protocols)

2004-01 *Thom Frühwirth, Marc Meister (eds.)*
First Workshop on Constraint Handling Rules

2004-02 *Christian Heinlein*
Concept and Implementation of C+++, an Extension of C++ to Support User-Defined Operator Symbols and Control Structures

2004-03 *Susanne Biundo, Thom Frühwirth, Günther Palm(eds.)*
Poster Proceedings of the 27th Annual German Conference on Artificial Intelligence

2005-01 *Armin Wolf, Thom Frühwirth, Marc Meister (eds.)*
19th Workshop on (Constraint) Logic Programming

2005-02 *Wolfgang Lindner (Hg.), Universität Ulm , Christopher Wolf (Hg.) KU Leuven*
2. Krypto-Tag – Workshop über Kryptographie, Universität Ulm

2005-03 *Walter Guttmann, Markus Maucher*
Constrained Ordering

2006-01 *Stefan Sarstedt*
Model-Driven Development with ACTIVECHARTS, Tutorial

2006-02 *Alexander Raschke, Ramin Tavakoli Kolagari*
Ein experimenteller Vergleich zwischen einer plan-getriebenen und einer leichtgewichtigen Entwicklungsmethode zur Spezifikation von eingebetteten Systemen

2006-03 *Jens Kohlmeyer, Alexander Raschke, Ramin Tavakoli Kolagari*
Eine qualitative Untersuchung zur Produktlinien-Integration über Organisationsgrenzen hinweg

2006-04 *Thorsten Liebig*
Reasoning with OWL - System Support and Insights –

2008-01 *H.A. Kestler, J. Messner, A. Müller, R. Schuler*
On the complexity of intersecting multiple circles for graphical display

2008-02 *Manfred Reichert, Peter Dadam, Martin Jurisch,l Ulrich Kreher, Kevin Göser, Markus Lauer*
Architectural Design of Flexible Process Management Technology

2008-03 *Frank Raiser*
Semi-Automatic Generation of CHR Solvers from Global Constraint Automata

2008-04 *Ramin Tavakoli Kolagari, Alexander Raschke, Matthias Schneiderhan, Ian Alexander*
Entscheidungsdokumentation bei der Entwicklung innovativer Systeme für produktlinien-basierte Entwicklungsprozesse

2008-05 *Markus Kalb, Claudia Dittrich, Peter Dadam*
Support of Relationships Among Moving Objects on Networks

2008-06 *Matthias Frank, Frank Kargl, Burkhard Stiller (Hg.)*
WMAN 2008 – KuVS Fachgespräch über Mobile Ad-hoc Netzwerke

2008-07 *M. Maucher, U. Schöning, H.A. Kestler*
An empirical assessment of local and population based search methods with different degrees of pseudorandomness

2008-08 *Henning Wunderlich*
Covers have structure

2008-09 *Karl-Heinz Niggl, Henning Wunderlich*
Implicit characterization of FPTIME and NC revisited

2008-10 *Henning Wunderlich*
On span-P^{cc} and related classes in structural communication complexity

2008-11 *M. Maucher, U. Schöning, H.A. Kestler*
On the different notions of pseudorandomness

2008-12 *Henning Wunderlich*
On Toda's Theorem in structural communication complexity

2008-13 *Manfred Reichert, Peter Dadam*
Realizing Adaptive Process-aware Information Systems with ADEPT2

2009-01 *Peter Dadam, Manfred Reichert*
The ADEPT Project: A Decade of Research and Development for Robust and Fexible Process Support
Challenges and Achievements

2009-02 *Peter Dadam, Manfred Reichert, Stefanie Rinderle-Ma, Kevin Göser, Ulrich Kreher, Martin Jurisch*
Von ADEPT zur AristaFlow® BPM Suite – Eine Vision wird Realität "Correctness by Construction" und flexible, robuste Ausführung von Unternehmensprozessen

2009-03 *Alena Hallerbach, Thomas Bauer, Manfred Reichert*
Correct Configuration of Process Variants in Provop

2009-04 *Martin Bader*
On Reversal and Transposition Medians

2009-05 *Barbara Weber, Andreas Lanz, Manfred Reichert*
Time Patterns for Process-aware Information Systems: A Pattern-based Analysis

Ulmer Informatik-Berichte
ISSN 0939-5091

Herausgeber:
Universität Ulm
Fakultät für Ingenieurwissenschaften und Informatik
89069 Ulm