Electronic Communications of the EASST
Volume 37 (2011)

Workshops der wissenschaftlichen Konferenz
Kommunikation in Verteilten Systemen 2011
(WowKiVS 2011)

Beyond Flexibility — Workflows in the perioperative Sector of the Healthcare Domain

Markus Bandt, Robert Kühn, Sebastian Schick, Holger Meyer

12 pages

Guest Editors: Horst Hellbrück, Norbert Luttenberger, Volker Turau
Managing Editors: Tiziana Margaria, Julia Padberg, Gabriele Taentzer
ECEASST Home Page: http://www.easst.org/eceasst/ ISSN 1863-2122

Beyond Flexibility — Workflows in the perioperative Sector of the Healthcare Domain

Markus Bandt[1], **Robert Kühn**[2], **Sebastian Schick**[1], **Holger Meyer**[1]

[1] {mb,schick,hme}@informatik.uni-rostock.de
Database Research Group
University of Rostock

[2] kuehn@it-science-center.de
IT Science Center Rügen gGmbH
Circus 14, 18581 Putbus

Abstract: The OR environment of a hospital demands high standards of both qualitative and economic management. Since the basic perioperative processes are rather well defined, the use of WfMS to support the planning and coordination of surgical treatment can be beneficial. The analysis of the perioperative work showed the need of flexibility at certain points in the processes. Within the project PERIKLES a corresponding YAWL model was implemented using several workflow flexibility concepts. A major challenge in this context was the handling of unpredictable exceptions (e.g. changing requirements during surgery). The use of exception handling is one reliable way to meet this requirements.

Keywords: Workflow, Flexibility, Healthcare, Perioperative process, YAWL, Exception handling

1 Introduction

Medical and especially clinical processes are known to be very complex and highly flexible in general. Among other reasons, this is caused by the wide variety of medical cases and treatments, anamnesis and individual progress of disease of a patient but also by the individual approach of work by the medical staff.

In clinical environments usually the operating room (OR) is the facility with the highest costs and revenues. As these processes typically bind a large number of expensive resources, coordination and scheduling aspects are of paramount importance. In our requirement analysis of the perioperative processes however we found that the general processes in this context are rather strictly defined. The implementation may differ in detail but all basic steps of the preparation and execution of surgical treatment can be combined in a generalised model. The task analysis was done using classical elements like work shadowing, interviews, and literature research [KBS+10]. The aim of the task analysis was to understand the flow of work in the perioperative process and to observe stakeholders and their tasks. After the task analysis we included the domain experts into the validation of the developed process models, which not only increased the acceptance of the results, but also eliminated ambiguities, inaccuracies and errors as soon as possible.

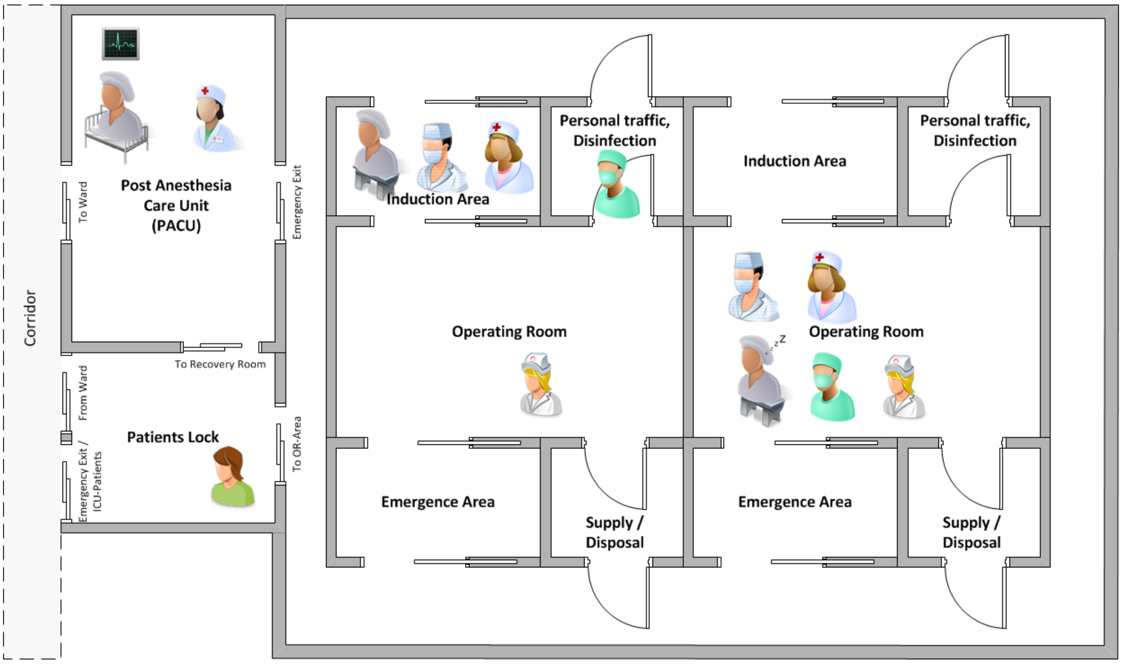

Figure 1: Operating area in a fictive hospital (adapted from [KBD+10])

1.1 The Perioperative Process at a Glance

The perioperative process itself includes all clinical workflows from admission of the patient on the ward, through the surgery, to care and patient release from hospital. To avoid confusion we consider in this article an operating area in a fictive hospital with only two ORs (Figure 1). Every OR has a separate induction and emergence area. Next to the induction is a room, which is used for personal traffic and for disinfection of the OR team. The room besides the emergence area is for supply or disposal of equipment, surgical inventory, medical instruments, and consumable supplies.

Normally, the stationary patient (inpatient) is brought to the patients lock by a staff nurse. Here the patient is transferred from his hospital bed onto the operating table. Afterwards the patient is anaesthetised in the induction room and the surgery takes place in the OR. After completion of surgery, the anaesthesia is abolished in the emergence room and the patient is transferred from the operating table to the hospital bed in the lock. The immediate care takes place in the post anaesthesia care unit (recovery room, PACU) and when the patients condition is stable, he will be brought to the ward again. Patients with a more critical status are carried over directly from the lock to the intensive care unit (ICU), where more intensive care is ensured. Figure 1 shows three different patient processes, one patient in the recovery room, one in the induction room, and one in the OR.

For any given patient case, many human (e.g. surgeons, anaesthesiologists, scrub nurses, technicians) and non-human resources (e.g. technical medical equipment, instrument sets, and capacity for post-operative care) are required. However, in general these resources are shared

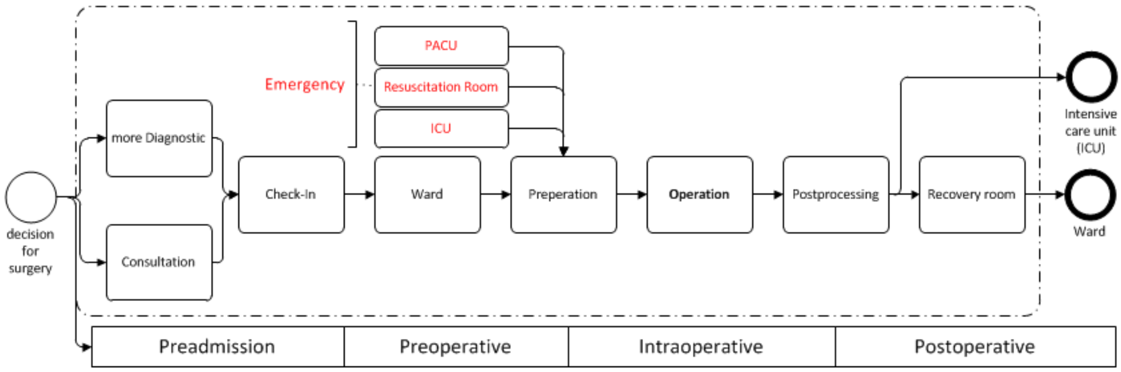

Figure 2: Perioperative process (adapted from [SGS03])

and scarce. Non-elective patients (urgency or emergency), staffing bottlenecks, and other uncertainties further complicate the OR coordination. Consequently, OR-centric workflows are characterised by the challenge to utilise these resources as efficiently as possible and to synchronize them with multiple concurrent, dynamic patient care processes. Generally the perioperative process can be divided into three sub-processes pre-, intra- and postoperative process, indicating the phases before, during, and after the surgery.

Besides the medical care of the patient, there are a number of complex administrative tasks to be done on the day of surgery. The latter is the job of the OR coordinator, who may be assisted by other decision-making people such as a charge nurse and an in-charge anaesthesiologist. While the operating unit generates the majority of the process costs, other departments such as the hospitals diagnostic centre, the ICU and the inpatient wards, are by no means negligible and must also be integrated into a cost-aware, process-oriented supporting solution.

On the whole, the complete surgery-related business processes must be well managed and scheduled in order to be cost-efficient while at the same time meeting the patients expectations of timely service delivery. These challenges call for a supporting system that is process-oriented, resource-centric, and schedule-aware.

PERIKLES aims at introducing an IT-based solution that meets these requirements. The project specifically focuses on the period of perioperative care which refers to both medical treatment as well as administrative tasks that take place before, during, or after the surgical procedure. To ensure optimal support for the planning and coordination of surgical procedures, we established an extended view of the traditional perioperative process. In PERIKLES the support of the OR management starts with the decision for surgery and ends with the patients transfer to the ward or to the ICU. Figure 2 shows simplified the phases of the expanded perioperative process and its main tasks.

- The **preadmission process** starts with the decision for surgery and includes the (preoperative and anaesthesiological) consultation and diagnostic steps.

- The **preoperative process** contains all tasks for planning the operation and preparing the patient.

- The **intraoperative process** consists of the surgery, which usually takes place in an OR or sometimes immediately in the ER (emergency or resuscitation room). The patient is, for example, transferred from the PACU or the ICU/ER to the OR. After the surgery the patient is transferred to the ICU or to the PACU.

- The **postoperative process** covers the postoperative care until the patients release from the hospital.

1.2 Terminology

In this paper we use the term *case* to refer to the medical case of a patient. The workflow instance of a medical case is named as *patient workflow*. In this context a *patient workflow* represents a logical surgical treatment (which itself may consist of several physical procedures) including all pre- and postoperative treatment (and care).

Concerning the patient we distinguish between *elective* and *non-elective* patients. The first class represents patients whose surgery is planned on the long-term (usually more than 24 hours). Elective cases can be subdivided into *inpatients* (patients who are admitted to the hospital and stay there till the primary treatment is finished) and *outpatients* (who just come to the hospital for treatment and leave it on the same day). An outpatient can become an inpatient due to medical reasons though, e.g. if the patients state of health is not as good as assumed.

For non-elective patients a surgery is unexpected and therefore cannot be planned at all. Beyond that we also make a distinction between *emergent* (emergencies) and *urgent* (urgencies) patients. An emergent patient must be operated as soon as possible. Urgent patients also require immediate actions, but due to the stable health state of the patient, it is possible to postpone the surgery for a short time.

The remainder of this paper is organised as follows. In Section 2 the flexibility problems are described and in Section 3 possible implementations are discussed. Section 4 gives an overview about related work. Section 5 gives a short summary and shows directions for the future work.

2 Need for Flexibility in the Perioperative Process

By its very nature, the perioperative process is highly dynamic. For instance, due to deterioration of his or her condition, an inpatient may become an emergency case and requiring immediate surgery. Unforeseen and unpredictable events are quite common and may call for ad-hoc re-coordination decisions, very often on the day of surgery itself. Therefore, a system assisting the OR manager must support the execution of business processes that can deviate from a predefined standard.

Given this scenario, our research initiative is investigating the potential of innovative software and hardware approaches to support typical coordination and cooperation challenges in a dynamic perioperative environment. Reasons for flexibility, dynamic change, and even ad-hoc task execution of the perioperative process are manifold: deviations, change, and conflicts in the patients and physicians schedule, equipment, and room availability. Additionally, the OR manager has to keep in mind certain restrictions during re-scheduling: surgical suite inventory bound to certain rooms, physician and surgical nurses privileges and specialities and interpersonal conflicts, to name some [MK07, MVK08].

We now will illustrate some situations or scenarios which happen every day in OR management. They will underscore the need for flexibility:

S1 The need to *re-schedule a single case* due to

- **S1.1** *change of surgical method* (e.g. non-invasive to invasive) due to unforeseen status of the operation area or need for different patient position
- **S1.2** *correction of diagnosis* or detection of additional diagnosis during surgical treatment can lead to additional or changed surgical procedures with different/additional physicians and surgical suite inventory

S2 *Predictable emergency situations* are very likely to happen one of the days. They have a strong impact and may cause re-scheduling of other cases, too:

- **S2.1** *emergency operations* of in- and outpatients
- **S2.2** *organ transplant* carried out by a special surgical team

S3 Furthermore, a lot of *unpredictable situations* have to be handled by the OR coordinator:

- **S3.1** *extended demand of additional resources* like an additional OR or additional staff members or a complete surgical team as impromptu reaction of change in the surgical procedure
- **S3.2** sudden *lack of resources* (e.g. staff, due to illness)
- **S3.3** *deterioration of the patients conditions* or flatline needing immediate response
- **S3.4** *disaster situation* like equipment malfunctions

There are also some situations within the perioperative process which are hard to model or even to be observed by the OP coordinator or tracked down by Auto-ID techniques. We will give some examples from the perioperative process to illustrate the need for special modelling concepts:

S4 The perioperative process has a lot of activities carried out in parallel. Only some activities are ordered, i.e., we have to model partial order of activities as shown in Figure 4.

S5 There are also many optional activities which can be skipped like the preoperative anaesthetic consultation which is handled differently for in- and outpatients.

S6 An example for alternative activities is the ICU patient who becomes temporary PACU patient due to lack of an ICU bed.

S7 The synchronisation of different workflows has to be expressed, e.g. a surgery can not start until surgical team and patient are in the right (OR) place.

S8 Even on the day of surgery the patient can withdraw the prior informed consent (PIC). This immediately leads to a cancellation of activities. And there are many other circumstances causing this as seen above, e.g. emergency operations postponing regular operations allowing procrastination.

This is only a small subset of re-scheduling events calling for flexibility in the perioperative process. We will now go on and describe which concepts can be used for the flexible modelling of the perioperative process.

3 Flexibility and Implementation within the Perioperative Process

To classify the concepts used within the PERIKLES project we will give a short overview about process flexibility. In literature [SMR+08, WSR09] you can find different taxonomies to classify flexibility within business process modelling and execution. Weber et al. using a process lifecycle model to present a taxonomy for dynamic processes (see Figure 3). The model distinguishes between four phases: *Design*, *Model*, *Execute* and *Monitor*. As depicted in Figure 3 flexibility types can occur in different phases of the process lifecycle. Techniques to implement flexibility concepts during the modelling phase are *granularity control*, *flexibility by enumeration*, *late binding* and *late modelling*. However the last two concepts are also applicable in the execution phase. Nevertheless runtime flexibility is a much more important challenge which deals with expected and unexpected exceptions. Expected exception will be handled with concepts like Exlets [AHAE07], which is provided by YAWL. Unexpected exceptions lead to ad-hoc changes with effects being instance-specific. Process schema evolution is one outcome of the monitoring phase.

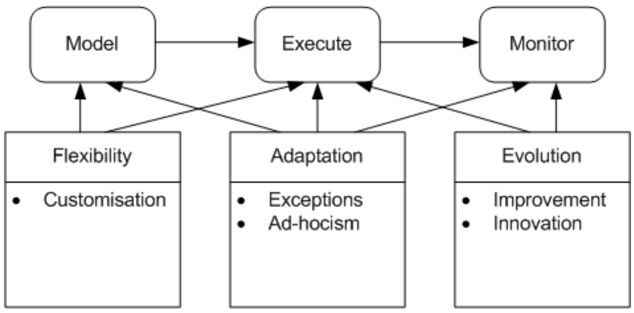

Figure 3: Taxonomy for dynamic processes [WSR09]

Besides this, Schonenberg et al. presents a slightly different survey about process flexibility in [SMR+08]. Their focus is on individual flexibility types. Therefore, four main types are presented. *Flexibility by design* implies techniques used within the modelling phase like alternative execution paths where parallelism, choice, iteration, interleaving, multiple instances or cancellation are possible realisation options. *Flexibility by deviation* allows to deviate the execution path at run time without altering the process model. Presented realisation options are undo, redo or skip of tasks, creation of additional instances of task and invoking task. *Flexibility by underspecification* combines the techniques of late binding and late modelling. *Flexibility by change* differentiates between change of process instances and change of the process model. In the latter case a change affects all new process instances.

3.1 Identified Modelling Concepts

To improve the traditional perioperative process we investigated existing process flexibility approaches in view of their applicability within PERIKLES context. One result of our case study is a catalog of techniques applicable for the PERIKLES scenario.

In PERIKLES the WfMS of choice was YAWL, because it uses the Workflow Patterns [AHKB03] as formal basement [HAAR10]. Moreover YAWL is open source and can be easily extended with new web services. Therefore, further considerations follow the provided concepts and capabilities of YAWL. We figured out different types of process flexibility used within perioperative process modelling task which can be assigned to the modelling and execution phase.

Regarding the project requirements we identified several structures applicable for modelling. Due to space restrictions we only present most appropriate solutions. However, every pattern is also realisable with other techniques.

C1 *Partial order* of tasks (see *S4*) describes the demand of task execution in a specified order. *Interleaved parallel routing* (Pattern 17 [AHKB03]) offers an adequate solution.

C2 *Optional tasks* (see *S5*) can be implemented using *deferred choice* (Pattern 16 [AHKB03]). Besides, *exclusive choice* (Pattern 4 [AHKB03]) combined with cancellation sets also offers a data independent implementation which should be used in case of optional subnets.

C3 *Iterations* of complex constructs (see *S6*) occur using optional or unconditional loops.

C4 *Alternative tasks* (see *S6*) are modelled with *deferred choice* (Pattern 16 [AHKB03]) as far as there is no need for subnets. In the latter case *exclusive choice* in combination with cancellation sets should be preferred. This is necessary because of YAWL engine implementation issues regarding the combination of *deferred choice* and subnets. In this case the subnet will be started regardless of whether other tasks are available.

C5 *Milestones* offer to execute tasks only if the process instance is in a specific state (see *S7*). This corresponds to Pattern 18 in [AHKB03].

However, besides structures within the modelling phase we also identified several structures which are applicable during execution phase. In contrast to the presented structures above, in some situations it is necessary to make changes to the executed process to treat uncertainty.

C6 *Skipping tasks* is a concepts which is recognized by our case study (see *S8*), but not supported by YAWL. Using Exception techniques is not applicable because exceptions are bound to specific tasks. Therefore, one exception for each task should be modelled. Using the skip operator of YAWL is not possible because the workitem have to be in state *allocated*. As PERIKLES is an assistance system this is also not realisable.

C7 *Cancellation of tasks* (see *S8*) will be handled using the exception service of YAWL. Exlets provide different actions on workitem or case level as presented in [AHAE07].

C8 *Expected and unexpected exceptions* (see *S8*) can also be handled using the exception service of YAWL (Exlets).

3.2 Implementation of the Identified Structures

As already mentioned we identified several generalised flexibility structures during the requirements analysis of the perioperative process. We will show the actual implementation of these structures in our workflow model at two examples.

In Figure 4 an example workflow for the **partial order of tasks** structure is shown. The induction of a patient consists of several tasks. The completion of these tasks is a prerequisite for the surgery. But the order is only partially fixed. The varying task is the *anaesthetic procedure* while the two event driven locating tasks *patient arrives at induction room*, *patient arrives at operating room* and the task *begin of surgery* are a fixed task sequence.

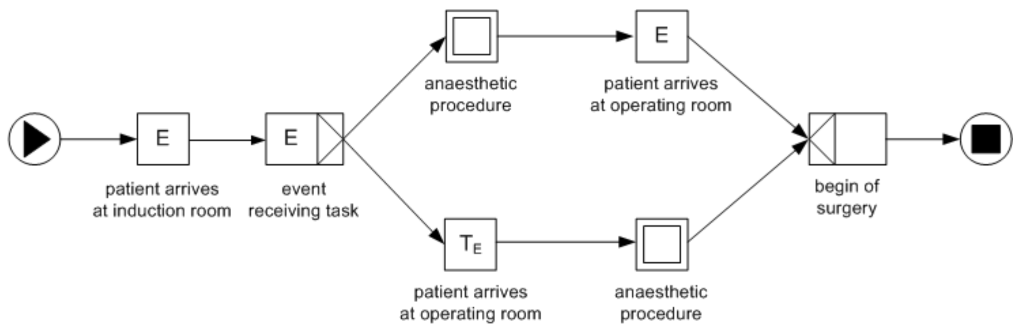

Figure 4: YAWL workflow of a patients induction

In general the patient will be narcotised in the induction room but for several medical reasons it can be necessary to execute the anaesthetic procedure not until the patient is located in the OR. For the OR coordinator it is important to know the exact state of the case at any given time. That is why we use a *XOR split* to distinguish between the two mutual ways of proceeding after the patient is located in the induction room.

The split itself is modelled as an *event receiving task* since the input is needed to decide which option shall be triggered. The locating task *patient arrives at operating room* will be triggered by an external Auto-ID event while the activation of the anaesthetic procedure would require a user generated event in our scenario.

As stated in Subsection 3.1 there are other concepts to integrate this structure in a model. In the results of our requirement analysis however we only discovered partial order structures with no more than one variable task and fixed sequences with three tasks at most. Considering this, the modelling and maintenance of our preferred solution is not very expensive compared to other approaches.

The second example shall illustrate how exception handling can be used to adapt a running patient workflow instance to predictable and unpredictable changes in the process. The main workflow in Figure 5 goes without annotation because of space limitation issues. The general idea however is, that task *A* stands for the preoperative sub-process while task *C* is a substitute

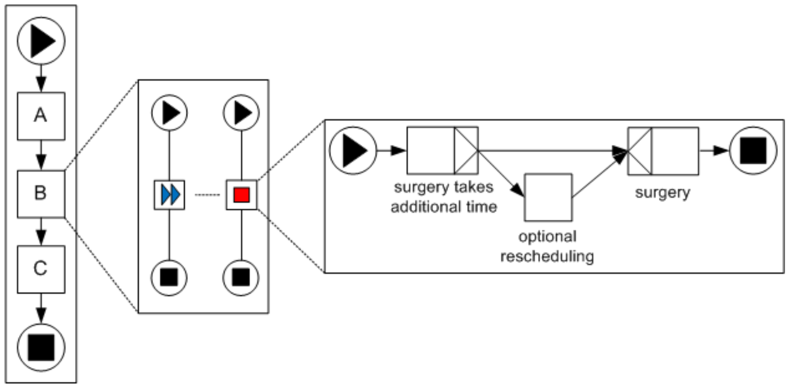

Figure 5: YAWL exception handling during surgery

for the postoperative sub-process in this representation. Consequential task *B* presents the actual surgery and is not a simple task but an Exlet.

During the course of the surgery the state of the patient or the runtime conditions of the task can change quite drastically. In case of such a change an exception is thrown and detected by the Exlet *(B)*. Depending on the type of the exception, specific actions will be performed.

In our example the surgery takes more time as planned which is a rather common situation. Though, if it is not the last surgery for that day in the corresponding OR, it may be necessary to reschedule the following cases. That is why a sub-process will be instantiated in the context of the Exlet enabling the OR coordinator to reschedule if needed.

Although in the example the handling of a deadline expiry event is shown there are several other types of exceptions which may require other actions to be handled properly. The full potential of exception handling with YAWL is presented in [HAAR10].

With these examples we showed how some of the identified flexibility structures were implemented in the context of the PERIKLES project. Due to space limitations for the other structures we will state the preferred concepts which were used in the PERIKLES model in tabular form only. In Table 1 we primarily refer to the Workflow Patterns as introduced in [AHKB03].

Table 1: Implementation of Flexibility Structures in the Perioperative Scope

	Flexibility structure	**Applicable concepts**	**Implementation**
C1	Partial order of tasks	Pattern 17	Non-parallel, Pattern 4
C2	Optional tasks	Patterns 16, 4^+	Pattern 16, 4^+
C3	Iteration of complex structures	optional, unconditional	optional, unconditional
C4	Alternative tasks	Patterns 16, 4^+	Pattern 16, 4^+
C5	Milestone based sync.	Pattern 18	Pattern 18
C6	Skip task	Exception handling	Selective exception handling
C7	Cancellation of tasks	Exception handling	Exception handling (Exlets)
C8	(Un-)Expected exceptions	Exception handling	Exception handling (Exlets)

$^+$ The use of cancellation sets is required.

4 Related Work

We describe in this paper the implementation of an assistance system for the perioperative process. Several work has been done in the area of supporting healthcare processes using workflow management systems but very few are using YAWL as workflow management system and especially were concerned with the perioperative process. Related work can be found in the general area of flexible business process management systems [MGR04, SMR+08, MRA+10]. Very few papers explore flexibility in workflows for healthcare, e.g. [RRVK10]. The article identifies several flexibility patterns but concentrates on the outpatient management in a Dutch hospital. Furthermore, how current workflow system would support such patterns is also part of the analysis. The pattern based approach is strongly related to the patterns presented in [MRA+10].

Müller, Greiner, and Rahm [MGR04] present a system called *AgentWork* providing support for automated workflow adaption. In the field of cancer chemotherapy the need of reactive and predictive adaptions of workflows is identified and a solution is presented. To cope with exceptions during workflow execution a ECA rule approach based on temporal logic was introduced. The event monitoring is described using ActiveTFL (Active Temporal Frame Logic) which is mapped to database triggers. Moreover, the approach is highly tied to data e.g. leukocyte count which is not applicable to the PERIKLES approach. Also it is difficult to describe different unforeseen exception as ECA rules.

In [MRA+10] the concept of Proclets is presented which deals with the gynaecological oncology healthcare process. They introduce autonomous processes that interact with each other via communication channels. Nevertheless, the presented approach is not yet applicable to real life process management within the perioperative process because "future work related to the verification, validation, and enactment of Proclets is necessary".

A lot of research is also done on operating room planning and scheduling systems. In [CDB10, BSW+10] e.g. different reviews where done which indicates the urgent need of effective planning and scheduling strategies. Besides this, different systems for planning and scheduling where presented. [ABC+09] presents a scheduling tool for hospital staff using concepts of expert systems with rules to generate the schedules. A visualisation system which allows for resource utilisation is presented in [BDC06]. The main focus of the system is on the detection of resource conflicts within the master surgery scheduling process. The drawback of these systems is the incapability of visualising online whether a resource is available or occupied during a surgery.

5 Conclusion

During the requirements analysis of the perioperative process we identified several recurring flexibility structures. In the course of the PERIKLES project we analysed various flexibility concepts and created a corresponding model implementing these structures.

Since PERIKLES utilises the WfMS YAWL as tracking system, we had to face certain limitations concerning some of the general workflow flexibility concepts. Considering this at some points in the model we might have favoured a stable and pragmatic solution over concepts with more potential for flexibility. However, by validating it with the stakeholders the model turned out to be sufficient for the perioperative process.

In PERIKLES the WfMS of choice was YAWL. It can be easily extended with new (web) services using a special interface of the engine. Within the project we added services for accessing external data sources, adding transactional support, and extended resource planning and management. Nevertheless, using different WfMS may improve abstraction and can lead to richer flexibility concepts. At least a cost benefit analysis has to be done in the future.

It turned out once more that one of the major challenges in the scope of OR management as well as emergency medicine and intensive care is the handling of unpredictable exceptions. In PERIKLES we used exception handling to adapt the patient workflow instance and keep the system in a consistent state. To completely meet up with the requirements of such situations the use of real-time ad-hoc processing and subsequent process mining seems promising. As it appears there is a need of further investigation on that matter.

Acknowledgements: Parts of this work were funded by the Bundesministerium für Bildung, Wissenschaft, Forschung und Technologie (BMBF), Bonn (Germany) under grant 01IS09009B and 01IS09009C.

Bibliography

[ABC+09] C. Ardito, P. Buono, M. Costabile, R. Lanzilotti, A. Simeone. An information visualization approach to hospital shifts scheduling. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5613 LNCS(PART 4):439–447, 2009.

[AHAE07] M. Adams, A. H. M. ter Hofstede, W. M. P. van der Aalst, D. Edmond. Dynamic, Extensible and Context-Aware Exception Handling for Workflows. In Meersman and Tari (eds.), *OTM Conferences (1)*. Lecture Notes in Computer Science 4803, pp. 95–112. Springer, 2007.

[AHKB03] W. van der Aalst, A. ter Hofstede, B. Kiepuszewski, A. Barros. Workflow Patterns. *Distributed and Parallel Databases* 14:5–51, 2003.

[BDC06] J. Beliën, E. Demeulemeester, B. Cardoen. Visualizing the demand for various resources as a function of the master surgery schedule: A case study. *Journal of Medical Systems* 30(5):343–350, 2006.

[BSW+10] A. Baumgart, G. Schüpfer, A. Welker, H.-J. Bender, A. Schleppers. Status quo and current trends of operating room management in Germany. *Current Opinion in Anaesthesiology* 23(2):193–200, 2010.

[CDB10] B. Cardoen, E. Demeulemeester, J. Beliën. Operating room planning and scheduling: A literature review. *European Journal of Operational Research* 201(3):921–932, 2010.

[HAAR10] A. H. M. Hofstede, W. M. P. Aalst, M. Adams, N. Russell (eds.). *Modern Business Process Automation YAWL and its Support Environment*. Springer Berlin Heidelberg, 2010.

[KBD+10] R. Kühn, M. Bandt, A. Dittmar, H. Meyer, P. Forbrig. HOPS zur Modellierung flexibler, klinischer Prozesse als Grundlage eines workflowbasierten Assistenzsystems. In *USEWARE 2010*. VDI-Berichte/VDI-Tagungsbände 2099, pp. 77–86. Baden-Baden, 13. und 14. Oktober 2010, 2010.

[KBS+10] R. Kühn, M. Bandt, S. Schick, I. Bruder, A. Heuer, P. Forbrig. Entwurf und Transformationskonzepte für flexible klinische Workflow Modelle. In Balke and Lofi (eds.), *Proceedings of the 22nd Workshop "Grundlagen von Datenbanken 2010" (GvD-2010), Bad Helmstedt, Germany, May 25-28, 2010*. Volume 581. 2010. ISSN: 1613-0073.

[MGR04] R. Müller, U. Greiner, E. Rahm. AGENTWORK: A workflow system supporting rule-based workflow adaptation. *Data and Knowledge Engineering* 51(2):223–256, 2004.

[MK07] R. A. Marjamaa, O. A. Kirvelä. Who is responsible for operating room management and how do we measure how well we do it? *Acta Anaesthesiologica Scandinavica* 51:809–814, 2007.

[MRA+10] R. S. Mans, N. C. Russell, W. M. P. van der Aalst, P. J. M. Bakker, A. J. Moleman, M. W. M. Jaspers. Proclets in healthcare. *Journal of Biomedical Informatics* 43(4):632 – 649, 2010.

[MVK08] R. Marjamaa, A. Vakkuri, O. Kirvelä. Operating room management: why, how and by whom? *Acta Anaesthesiologica Scandinavica* 52(5):596–600, 2008.

[RRVK10] H. A. Reijers, N. C. Russell, S. b. Van Der Geer, G. A. M. c. Krekels. Workflow for healthcare: A methodology for realizing flexible medical treatment processes. *Lecture Notes in Business Information Processing* 43 LNBIP:593–604, 2010.

[SGS03] W. S. Sandberg, T. J. Ganous, C. Steiner. Setting a Research Agenda for Perioperative Systems Design. *Surgical Innovation* 10(2):57–70, 2003.

[SMR+08] H. Schonenberg, R. S. Mans, N. Russell, N. Mulyar, W. M. P. van der Aalst. Process flexibility: A survey of contemporary approaches. *Lecture Notes in Business Information Processing* 10 LNBIP:16–30, 2008.

[WSR09] B. Weber, S. W. Sadiq, M. Reichert. Beyond rigidity - dynamic process lifecycle support. *Computer Science - R&D* 23(2):47–65, 2009.