In: Kaiser, K., Miksch, S., Tu, S.W. (eds.): Computer-based Support for Clinical Guidelines and Protocols. Proceedings of the Symposium on Computerized Guidelines and Protocols (CGP 2004). Prague. IOS Press. 2004. p. 113-117

Adaptive Guideline-based Treatment Workflows with AdaptFlow

Ulrike GREINER, Jan RAMSCH, Barbara HELLER,
Markus LÖFFLER, Robert MÜLLER, Erhard RAHM
University of Leipzig
{greiner, mueller, rahm}@informatik.uni-leipzig.de
{ramsch, heller, loeffler}@imise.uni-leipzig.de

Abstract. One goal in modern medicine is to increase the treatment quality. A major step towards this aim is to support the execution of standardized, guideline-based clinical protocols, which are used in many medical domains, e.g., for oncological chemotherapies. Standardized chemotherapy protocols contain detailed and structured therapy plans describing the single therapy steps (e.g., examinations or drug applications). Therefore, workflow management systems offer good support for these processes. However, the treatment of a particular patient often requires modifications due to unexpected infections, toxicities, or social factors. The modifications are described in the treatment protocol but not as part of the standard process. To be able to further execute the therapy workflows in case of exceptions running workflows have to be adapted dynamically. Furthermore, the physician should be supported by automated exception detection and decision support for derivation of necessary modifications. The AdaptFlow prototype offers the required support for the field of oncological chemotherapies by enhancing a workflow system with dynamic workflow adaptation and rule based decision support for exception detection and handling.

Introduction

One goal in modern medicine is to increase treatment quality. A major step towards this aim is the use of standardized, guideline-based clinical protocols, which are used in many medical domains, e.g., for oncological chemotherapies. A chemotherapy treatment protocol contains detailed plans for diagnosis, therapy, toxicity management, and follow-up (compare e.g. [1],[2]). It additionally specifies how to deal with exceptional situations, which may often occur during therapy execution (e.g., an infection or a toxicity) and require a modification of the treatment of a particular patient so leading to a significant flexibility of treatment processes.

To adequately support the execution of guideline-based therapies and to relief the medical staff (even a specialized physician has to administer many different protocols to different patients at the same time) a system is needed that handles well-structured but flexible therapy processes efficiently. A convenient system should observe the status of the therapies currently being applied, offer automatic recognition of exceptional situations and appropriate decision support for handling such situations. Furthermore, the system should be able to automatically adapt affected therapy processes to adequately handle necessary treatment modifications. For some of these problems, e.g., the handling of data and knowledge or the organization of clinical activities, there are already several systems in use in the medical domain ([3],[4],[5],[6]). But to meet all requirements, a hybrid approach is necessary that combines a workflow management system with a rule base. The workflow system is used for the execution of therapy workflows and the integration of different users, data and applications. Furthermore, it has to support dynamic adaptation of running workflows to handle the flexibility of therapy processes ([7],[8]). The rule base handles the medical knowledge represented in the protocols and is used to detect exceptional situations.

The AdaptFlow prototype presented in this paper implements this hybrid approach and offers the following advanced features to support adaptive guideline-based treatment workflows. Besides *rule based exception detection* with extended Event-Condition-Action (ECA) rules and *dynamic workflow adaptation* with obligate user confirmation (i.e., if the dynamic adaptation is not appropriate the user can perform an alternative exception handling manually) it supports *two strategies for automatic workflow adaptation*. Whenever possible AdaptFlow tries to predictively adapt the remaining part of a running therapy workflow as soon as an exception is detected. This informs users early about necessary changes and supports a timely and effective treatment of patients. This strategy is based on temporal estimates about the duration of future workflow parts, which are not always possible. In those cases the second strategy, reactive adaptation, is used which adapts affected activities directly before their execution.

In the following we provide an overview of AdaptFlow.

1. AdaptFlow

We first present the architecture of AdaptFlow and then explain its rule-based exception handling and the strategies used for automatic workflow adaptation.

1.1 Architecture

Figure 1 shows the architecture of the AdaptFlow prototype consisting of a workflow management system, monitoring and adaptation modules, a rule base and a patient database. Since we wanted to avoid the development of a completely new workflow system we first evaluated whether we could extend an existing system. This was not feasible for commercial workflow systems since they still do not support adaptation of running workflow instances. However, there are research prototypes supporting workflow adaptations. We could use one of them as a basis for our extensions, namely ADEPT$_{flex}$ workflow system ([9]). It provides manual workflow adaptation and a Java Application Programming Interface (API) which is used by our extensions to perform automatic workflow adaptations. Furthermore, it offers a workflow definition language in which the protocols can be represented.

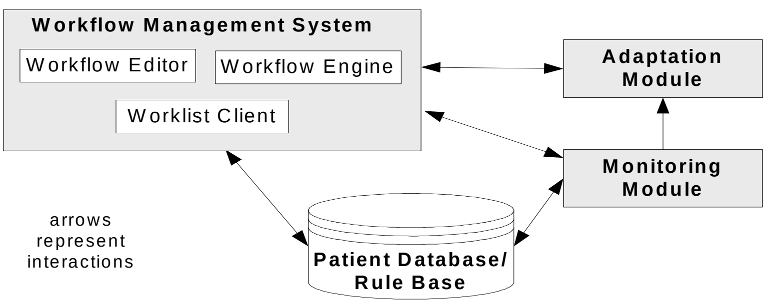

Figure 1. AdaptFlow architecture.

The workflow system includes an editor for workflow modelling, a workflow engine for workflow execution and worklist clients for the presentation of the activities to the users. The patient database contains the relevant information about the treated patients; it is a relational database and is accessed by workflow activities and users. If an existing patient database is to be used, tables, triggers and stored procedures will have to be added to offer the functionality required by AdaptFlow.

The exceptions supported by the system are maintained in the rule base. We use extended ECA rules (see below) to specify exceptional events and their handling. Currently, triggers and stored procedures on the patient database implement most rules. For instance, we use triggers to check whether newly inserted or updated data such as new laboratory measurements exceed some threshold so that therapy modifications may become necessary. Stored procedures are used to take additional data into account during rule evaluation. The monitoring module passes the derived reactions specified by an ECA rule to the adaptation module, which executes corresponding workflow adaptations. Each adaptation has to be confirmed by an authorized medical user before the workflow is actually changed. The user may reject a suggested workflow adaptation and perform an appropriate manual exception handling, e.g. a manual workflow adaptation.

1.2 Exception detection

Different oncological protocols were analyzed to find the exceptional events that may require treatment adaptations. Three main event types were identified: *medical events* (e.g., laboratory or pathological findings, diagnosis or toxicological ratings), *organizational events* (e.g., resource conflicts of medical devices, change of patient status due to hospitalization), and *social events* (e.g., loss of patient compliance for the treatment). Most of these events may only lead to exceptions in certain parts of the treatment. For instance, a low white blood count (in an acceptable range) may be tolerable after a cytostatika application phase, but not during other treatment phases. This illustrates the need to consider temporal validity intervals for exception handling.

To detect exceptional events it is necessary to establish rules that represent the conditions described in the treatment protocols. AdaptFlow uses extended Event-Condition-Action (ECA) rules of the following format (see [10] for more details):

WHEN	*exceptional event*	*WHEN*	*new finding for patient P*
WITH	*condition*	*WITH*	*white blood count < 1000*
THEN	*treatment adaptation*	*THEN*	*delete applications of drug A for P*
VALID-TIME	*time period*	*VALID-TIME*	*during the next seven days*

The event-condition-part (WHEN/WITH) specifies the event and the conditions under which treatment adaptations are required. The action part (THEN) describes the necessary treatment modifications. The optional valid time part (VALID-TIME) specifies a time period during which the reaction should be applied. It is either denoted by a fixed time interval or date, or by a conditional time interval whose end is specified by a condition that has to be satisfied (e.g., until the blood value has improved). This extension of ECA rules with temporal information is especially needed to meet medical requirements formulated in the guidelines. The sample ECA rule shown above on the right side specifies that for a patient with a critical blood value all applications of drug A should be dropped during the next seven days (syntax simplified for better readability).

The following adaptations of medical treatments can be specified in the action part of ECA rules: start, stop, and abort treatment; postpone, substitute, add, and delete treatment part; change properties of treatment and treatment part. A treatment modification requires one or more workflow adaptations. For instance, the deletion of a therapy part may result in dropping applications of different drugs. The necessary workflow adaptations for a particular treatment modification are determined by the monitoring module and specified with so called *control actions*. Control actions describe the workflow adaptations on the basis of activities, i.e. single workflow tasks. Each control action consists of the description of the workflow adaptation and the valid time as specified in the ECA rule. AdaptFlow supports the following control actions (W denotes a workflow; A,B activities; P a patient; t a time interval; p a parameter; f a function):

- dropping an activity <drop(A,P)>,
- adding a new activity <add(A,P)>,
- replacing an activity by another one <replace(A,B,P)>,
- delaying an activity <postpone(A,t,P)>,
- changing input parameters of an activity <change-value(A,p,f,P)>,
- aborting a workflow instance <abort(W,P)>,
- suspending a workflow instance <suspend(W,t,P)>.

For instance, the ECA rule shown above is translated into the control action *drop(application_of_drug_A, P)@[for next 7 days]* (the '@' separates the adaptation description from the valid time part).

1.3 Workflow adaptation

The implementation of the control actions either follows a predictive or reactive adaptation strategy. If possible, workflows should be adapted predictively, i.e. as soon as an exception is detected, in order to inform workflow users in time about the changed situation.

Predictive adaptation is possible if the valid time VT of a control action is specified by a fixed (absolute) date or interval. In this case, AdaptFlow can estimate the part of the workflow that will presumably be executed during VT and adapt it in advance. The estimation is based on the average execution duration of activities and other workflow constructs and is determined for each path through the remaining part of a running workflow that is affected by an exception. The expected duration of activities or time intervals between two activities can be specified during workflow definition; in the future we want to additionally use measured execution times. A detailed description of workflow estimation taking into account parallel and conditional paths and other workflow constructs can be found in [10].

Then, the control action specified in the ECA rule is applied to all affected activities in the workflow part corresponding to the valid time interval VT. For this purpose the API functions of the workflow system are used, e.g. to drop or add nodes. The workflow fragments in figure 2 illustrate the result of the application of the drop

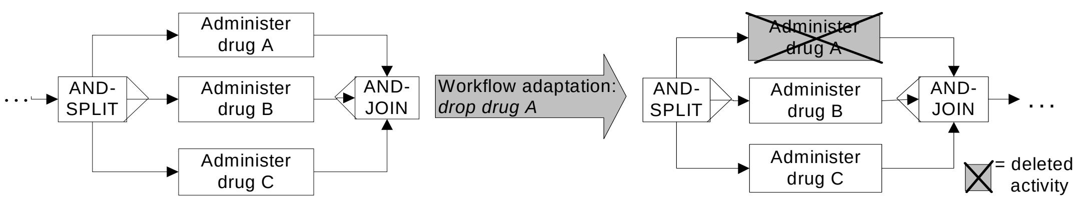

Figure 2. Workflow adaptation.

control action introduced above to a therapy workflow. The adaptations are shown to an authorized medical user for confirmation before the modified workflow instance is resumed. The predictive approach thus informs the user timely about necessary changes, in particular to prepare new activities (e.g., drug administrations) or cancel the preparation of deleted activities.

Reactive adaptation is used if the valid time VT of a control action is denoted by a conditional time interval (e.g., until the blood value has improved), or if the workflow contains conditional parts such as conditional splits or loops. In these cases, no estimation of the remaining workflow part is possible as it is not known in advance when the specified condition will be satisfied. Therefore, AdaptFlow monitors the further execution of running workflows that are affected by the exception until the condition is satisfied. Each affected activity is adapted according to the control action directly before its execution. After a user has confirmed the adaptations the workflow execution is continued.

2. Conclusion

The AdaptFlow prototype offers support of adaptive guideline-based treatment workflows by handling exceptions occurring during runtime widely automatically. It enhances a workflow system with automated exception detection, derivation of necessary workflow adaptations and automatic application of the workflow adaptations. Furthermore, AdaptFlow offers a predictive and a reactive adaptation strategy. We expect that AdaptFlow's automatic exception detection and handling will help increase treatment quality by enhancing the protocol conformance of the applied treatments and facilitate the treatment application for the medical staff.

Acknowledgement

This work has been supported by the German Research Association under grant Ra 497-12. $\text{ADEPT}_{\text{flex}}$ has kindly been provided by the Dept. of Computer Science, University of Ulm. We would also like to thank the medical staff of the University Hospital Leipzig and the staff members of the Coordination Centre for Clinical Trials Leipzig for helpful discussions.

References

[1] Schmitz N, Glass B, Kloess M, Loeffler M. Mega-CHOEP Phase III, DSHNHL 2002-1, 27.02.2003. Available at: http://www.lymphome.de/Gruppen/DSHNHL/Protokolle/2002-1/index.jsp accessed 2003/10/2.

[2] Hahn EG, Hohenberger W. InTACT: Interdisciplinary Trial for Adjuvant Colon Cancer Treatment (UICC III). Friedrich-Alexander-University; Erlangen-Nuremberg, Germany; 1997.

[3] van Bemmel JH, Musen M. editors. Handbook of Medical Informatics. Heidelberg: Springer; 1997.

[4] Degoulet P, Fieschi M. Introduction to Clinical Informatics. Springer; 1999.

[5] Quaglini S, Stefanelli M, Lanzola G, Caporusso V, Panzarasa S. Flexible guideline-based patient careflow systems. Artificial Intelligence in Medicine 2001; 22(1): 65-80.

[6] Musen M, Tu SW, Das AK, Shahar Y. EON: A Component-Based Architecture for Automation of Protocol-Directed Therapy. Journal of the American Medical Informatics Association 1996; 3(6): 367-388.

[7] Dadam P, Reichert M, Kuhn K. Clinical workflows - the killer application for process-oriented information systems? In: Proc. 4th Int. Conf. on Business Information Systems, 2000. Berlin: Springer; 2000; 36-59.

[8] Miksch S, Kosara R, Seyfang A. Is Workflow Management Appropriate for Therapy Planning? In Proc. of EWGLP 2000. Amsterdam: IOS Press; 2001: 53-69.

[9] Reichert M, Dadam P. $\text{ADEPT}_{\text{flex}}$ - supporting dynamic changes of workflows without losing control. Journal of Intelligent Information Systems 1998; 10: 93-129.

[10] Müller R. Event-Oriented Dynamic Adaptation of Workflows: Model, Architecture and Implementation. PhD Thesis; University of Leipzig; 2002.