*The New Navigators: from Professionals to Patients*
*R. Baud et al. (Eds.)*
*IOS Press, 2003*


# A UML Approach to Process Modelling of Clinical Practice Guidelines for Enactment

**T. Knape[a], L. Hederman[a], V. P. Wade[a], M. Gargan[a], C. Harris[a], Y. Rahman[b]**

*[a] Department of Computer Science, Trinity College, Dublin, Ireland.*
*[b] Diabetes Day Care Centre, St. James's Hospital, Dublin, Ireland.*

**Abstract**

*Although clinical practice guidelines (CPGs) have been suggested as a means of encapsulating best practice in evidence-based medical treatment, their usage in clinical environments has been disappointing. Criticisms of guideline representations have been that they are predominantly narrative and are difficult to incorporate into clinical information systems.*
*This paper analyses the use of UML process modelling techniques for guideline representation and proposes the automated generation of executable guidelines using XMI. This hybrid UML–XMI approach provides flexible authoring of guideline decision and control structures whilst integrating appropriate data flow. It also uses an open XMI standard interface to allow the use of authoring tools and process control systems from multiple vendors.*
*The paper first surveys CPG modelling formalisms followed by a brief introduction to process modelling in UML. Furthermore, the modelling of CPGs in UML is presented leading to a case study of encoding a diabetes mellitus CPG using UML.*

***Keywords:***
Clinical Practice Guidelines; UML; Workflow; Process Modelling; Enactment; Medical Informatics

## 1 Introduction

Clinical Practice Guidelines (CPGs) contain clinical knowledge that is used to ensure and improve quality of healthcare, to reduce inappropriate variations in clinical practice and healthcare costs, used for medical education, alerts and reminders, case management and decision support. Unfortunately many guidelines may not be used to their full extent since they are in a narrative format on paper or electronic file and therefore difficult to incorporate into clinical practice. Information technology (IT) may support processes of clinical care by enacting guidelines, integrating them with patient records and clinical care systems.

In literature one of the main goals of guideline modelling formalisms (e.g. GLIF, EON, ...) is the development of clinical decision support systems [5;6]. The approach presented in this paper aims at specifying CPGs for a later enactment in a workflow[1] management system (WfMS). CPGs can be considered as processes [8] which is what WfMS enact. Processes are sets of partially ordered steps to reach a goal [11]. Enactment is the execution of process steps according to a process definition. A CPG, which is in a form that can be "executed", is thus called an enactable CPG.

Mostly guidelines for enactment have been represented with specially designed guideline modelling formalisms. The enactment then requires a specially built execution engine that understands the guideline formalism. An alternative approach is to use mainstream IT business process modelling formalisms to represent guidelines so that they can be enacted using mainstream (workflow) engines. This has the potential to leverage developments in workflow technology, which are likely to be more significant than

[1] Workflows are business processes in execution (instances of a process model) in a computing environment [3].

developments in specific guideline enactment tools. This paper describes research into the use of UML to represent guidelines for enactment in a WfMS environment.

Two main phases of development can be identified in the process to enact a narrative CPG in a WfMS. The first phase concentrates on creating formal guideline process definitions of a CPG whereas the second phase concentrates on the actual enactment of the formal process definition in a WfMS. A prototype capturing all aspects of the two phases has been developed and demonstrated in a workshop at the Seventh Annual Conference of the Healthcare Informatics Society of Ireland. This paper focuses on the first phase depicted in Figure 1.

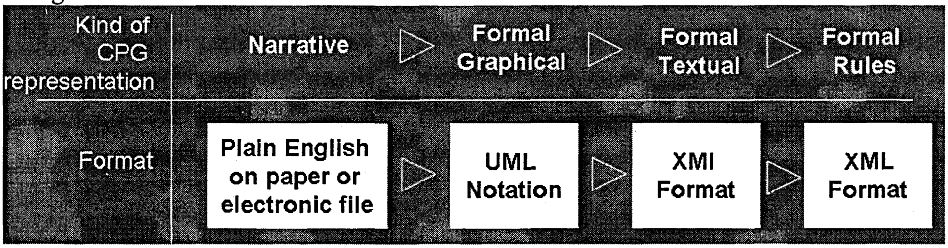


Figure 1: Different stages in modelling enactable CPGs

The transformation of a CPG represented in narrative form to a first formal human-readable graphical representation involves human intelligence and therefore may not be automated. Both the transformation from formal graphical to formal textual and from formal textual to formal rules can be automated, as outlined in section 6.

The paper first discusses modelling formalisms of CPGs. Commonalities among formalisms will be presented. Process modelling in UML is presented in section 3 followed by a discussion on how CPGs may be modelled using UML in section 4. A case study of a CPG on diabetes mellitus demonstrates the findings.

## 2 CPG Modelling Formalisms

Wang et al. [10] surveyed eleven CPG representation models to determine the modelling primitives and constructs which appear to be necessary for guideline representation. Each of the models surveyed had primitives for representing *decisions* and *actions*. Decisions select, automatically or through user interaction, from a set of alternatives, e.g. selection of a medication from a set of potential medications. Actions can be clinical interventions, data collections and wait states recommended by the guideline, e.g. take a chest x-ray. Most of the models surveyed had a *patient state* primitive; others had an *execution state*. Wang et al. contend that these two primitive types are two sides of the same coin. Patient states describe the state of a patient as a result of an intervention or decision (e.g. patient has diabetes mellitus). Execution states describe the status of the guideline process and its actions (e.g. the guideline process is *ready* to start or evaluation task of treatment options for the diabetic patient is *finished*). All the models surveyed provided some way of specifying *scheduling constraints* on the primitives mentioned above. Scheduling constraints are sequential or concurrent ordering of primitives (usually a combination of both). To reduce the level of complexity most guideline representation models support *nesting* which breaks a comprehensive and complicated guideline into several subguidelines creating multiple levels of abstraction. A subguideline is represented as a composite action on a higher level.

The next section presents UML features for process modelling. Following that the UML features are associated with guideline primitives presented above.

## 3 Process modelling in UML

UML was initially conceived as a general-purpose language for modelling object-oriented software applications [7]. Today it is considered as the lingua franca in software engineering. It is frequently used to illustrate processes in software applications.

Process modelling in UML is supported through activity diagrams. Notational elements of activity diagrams are activities, which are placeholders for software components carrying out business functions. The order of the flow of activities is determined by transitions, forking and synchronisation elements while the direction of the flow of activities is determined by decision diamonds. These together comprise control flow. (See figure 2; solid arrows represent control flow transitions; decisions are diamonds and forks and synchronisation are expressed by solid bars)

To enact a process, a WfMS needs to know which activity to call next and what data the activity needs. Thus control flow must be augmented with data flow. Data flow is the connection of data objects with activities that require them as input or / and produce them as output[2]. Data objects which are the subject of manipulation by activies in workflow processes are often called cases. Data flow is illustrated in Figure 2 by dashed arrows. The internal structure of data objects is described in class diagrams which in contrast to activity diagrams (dynamic view) represent the static view to the system. Figure 3 shows a small example of a patient data model with the patient class containing the patient ID (PatID) and a 'TestResult' class as a specialisation of the patient class containing the test name and test result as attributes.

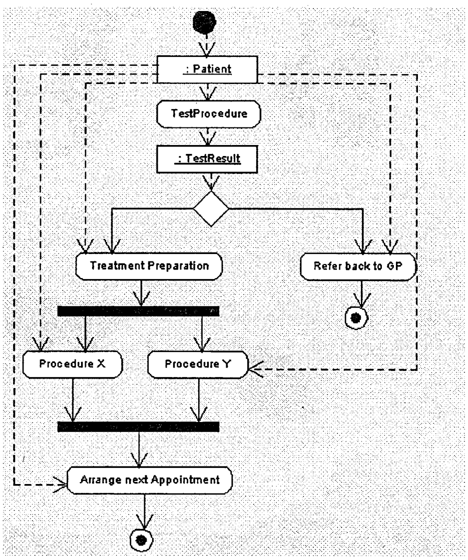


Figure 2: Activity Diagram for a Clinical Process

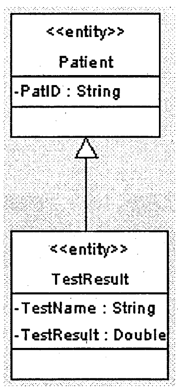


Figure 3: Class Diagram showing data structure of a workflow case

The concept of states is very well supported by activity diagrams since they are based on state diagrams. Every node of an activity diagram is a state, which may have e.g. an activity attached to itself. Each data object may contain state information describing the object closer. Every activity diagram starts with an initial state and ends with (a) final state(s). Activities contain different internal states that describe the progress of the activity.

[2] The concept of data flow is described as "object flow" in UML [9].

Activity diagrams may also be hierarchically split into various sub-diagrams reducing the level of complexity on higher levels of abstractions.

The following section discusses how the different aspects of CPG modelling formalisms (see section 2) may be modelled using UML.

## 4 Representation of CPGs in UML

Both, CPG modelling formalisms and UML have similar artefacts, which may be used to model clinical processes. As mentioned above process modelling in UML is best accomplished using activity and class diagrams. The following will thus briefly identify corresponding modelling artefacts of both approaches and section 5 will illustrate their use in a practical example.

Action steps, sometimes also called tasks in CPG terminology, may be modelled as activities in activity diagrams. CPG decision steps appear to be richer constructs than UML decision diamonds. For example, in GLIF[3] [4] the decision element is called a decision step. Decision steps can be either case steps (automatable) or choice steps. While case steps correspond to UML decision diamonds, choice steps incorporate an interaction with a clinician. Choice steps can be modelled in UML by combining an activity, accounting for the interaction with the clinician, with a decision diamond [2].

Scheduling constraints can be modelled by combining elements like transitions, forks and synchronisation. Nesting is also supported in UML through hierarchical sub diagrams.

Patient states can be supported in UML by adding a state to a data object. Data objects may be patient objects. Execution states are well documented in UML. They range from initial and final states indicating if the process has just started or is finished to states contained in activities.

Furthermore, UML supports stereotyping through its extension mechanism which allows the graphical notation to be tailored for clinicians by introducing easy to understand elements that map to complex combinations of UML elements.

The following section illustrates the use of process modelling elements for CPG representation in UML by example.

## 5 Case Study – A Guideline for Diabetes Mellitus

For our research we have chosen to encode the CPG on diabetes mellitus which is currently used in St. James Hospital Dublin. It is based on a simple narrative description and experience of healthcare professionals.

The enactable guideline process model comprises three different activity diagrams and one class diagram as the data model. The first guideline ("Guideline of Guidelines", see Figure 4.) acts as a coordination instance and directs the flow of activities to either the diabetes management or the diabetes diagnosis nested guideline. The guideline on diagnosis does not have to be executed for each patient since patients may already be diagnosed and just need treatment.

Because the MediLink environment, in which these CPGs are to be enacted, includes a Synapses federated patient records server [1], we leave it to each activity to access the patient data it needs, given the Patient ID (PatID). Thus we only need to pass PatIDs to most activities (see Figure 4) reducing the volume of data passed between the activities.

The process model starts with an initial state leading into the patient data object containing the PatID. The patient data object has outgoing data flow transitions to all activities modelled in the guideline process. The first activity in the guideline ("diabetesQuery") connects to the EHCR and requests information about the diagnosis

[3] A study on how GLIF3 can be represented in UML is given in [2].

status of the patient. The return value is stored in the data object "queryResult". Depending on the status of diagnosis of the patient (either diagnosed or not diagnosed) the following decision diamond automatically directs the flow of activities. The activity can be either "diabetesManagement" or "diabetesDiagnosis". Both activities can modify the patient's EHCR since they get the PatID as an input parameter. When "diabetesDiagnosis" is finished the next activity "diabetesReQuery" accesses the patient's EHCR again and requests information about the current status of diabetes mellitus. The request result is

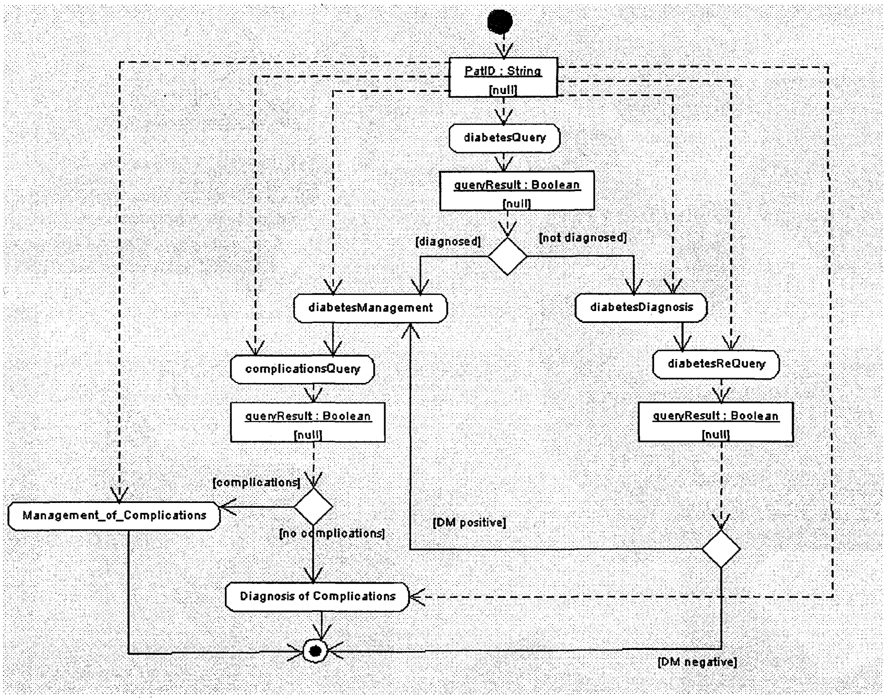


Figure 4: Guidelines of Guidelines

again stored in the next data object "queryResult". Depending on the data in this data object, the decision diamond directs one of the two outgoing branches. If the patient is still negative the guideline will end here, otherwise "diabetesManagement" is invoked as the next activity. The rest of the guideline is similar to the remarks made about the first half of the guideline.

## 6 Automating the UML guideline

To enact a guideline in our workflow engine Serene, the UML activity diagram is converted into workflow process rules. First the graphical UML is exported to XMI, a textual representation based on XML. The XMI is then converted to workflow rules using a converter, which was developed using XSLT (a programmable bridging mechanism between different XML documents). Then the activities in the process are bound to components, at which stage the process can be enacted using the workflow engine. A tool is under development to allow activities to be matched to the blueprints of methods of the business components in the earlier phase of graphical modelling

## 7 Conclusions and Further Work

This paper has discussed how knowledge, represented in CPGs, may be used to model clinical processes in the mainstream modelling language UML. Commonalities of domain-specific CPG modelling approaches (e.g. action steps, decision steps, patient states, execution states, and scheduling constraints) were outlined and a brief overview of process modelling in UML was presented.

Using mainstream technologies (UML, XMI, workflow), as proposed in this paper, may have to following potential benefits:

- vendor independence since every UML modelling tool exports its model in XMI format
- benefits from further innovative developments of the used technologies
- standardised technologies may make guidelines shareable when using the same technology

In the MediLink research programme work is currently ongoing to extend UML to make it easier to use for non-technical users, e.g. clinicians.

### Acknowledgements

The research reported in this paper is part of the MediLink Programme in Health Informatics, which is funded by the Programme of Research in Third Level Institutions of the Higher Education Authority in Ireland. Their support is gratefully acknowledged.

**Address for correspondence**

Correspondence and requests for materials should be addressed to Thomas Knape, email: thomas.knape@cs.tcd.ie.

**References**

[1] Grimson W, Berry D, Grimson J, Stephens G, Felton E, Given P, O'Moore R. Federated healthcare record server--the Synapses paradigm. Int J Med Inf 52: pp. 3-27, 1998.

[2] Hederman, L., Smutek, D., Wade, V., and Knape, T. Representing Clinical Guidelines in UML: A Comparative Study. G. Surján et al. Health Data in the Information Society, Proceedings of MIE2002. pp. 471-477. Amsterdam , IOS Press, 2002.

[3] Leymann F, Roller D. Production workflow concepts and techniques. Upper Saddle River, N.J, London: Prentice Hall PTR. Prentice-Hall International, 2000.

[4] Peleg M, Boxwala AA, Ogunyemi O, Zeng Q, Tu S, Lacson R, Bernstam E, Ash N, Mork P, Ohno-Machado L, Shortliffe EH, Greenes RA. GLIF3: the evolution of a guideline representation format. Proc AMIA Symp, pp. 645-9, 2000.

[5] Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R. A., Hall, R., Johnson, P. D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E. H., and Stefanelli, M. Comparing Models of Data and Knowledge for Guideline-Based Decision Support: a Case-Study Approach (Part 2 of 2). 2002.

[6] Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R. A., Hall, R., Johnson, P. D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E. H., and Stefanelli, M. Comparing Models of Decision and Action for Guideline-Based Decision Support: a Case-Study Approach (Part 1 of 2). 2002.

[7] Selic B, Ramackers G, Kobryn C. Evolution, not revolution . Communications of the ACM 45: pp. 70—72, 2002.

[8] Tu, S. W., Johnson, P. D., and Musen, M. A. A Typology for Modeling Processes in Clinical Guidelines and Protocols. AMIA Annual Symposium. 2001. San Antonio.

[9] UML Revision Taskforce. OMG UML Specification v. 1.4. 2001. Object Management Group.

[10] Wang D, Peleg M, Tu SW, Boxwala AA, Greenes RA, Patel VL, Shortliffe EH. Representation primitives, process models and patient data in computer- interpretable clinical practice guidelines: A literature review of guideline representation models. Int J Med Inf 68: pp. 59-70, 2002.

[11] Westfechtel B. Models and tools for managing development processes. Berlin, London: Springer, 1999.