2nd International Workshop on

Process-oriented information systems in healthcare

ProHealth '08

In conjunction with 6th Int'l Conf. on Business Process Management (BPM 2008), Milan, Italy

Co-Chairs:

Richard Lenz[1], Mor Peleg[2], Manfred Reichert[3]

[1] Department of Computer Science 6 (Data Management),
Friedrich-Alexander-University Erlangen-Nuremberg, Germany
[2] Department of Management Information Systems, University of Haifa, Israel
[3] Institute for Databases and Information Systems, University of Ulm, Germany

Process mining in healthcare: a contribution to change the culture of blame

- invited talk -

Silvana Quaglini

Department of Computer Science and Systems, University of Pavia, Italy
silvana.quaglini@gmail.com

Abstract. Something is changing in the medical culture. The diffusion of the statistics on medical errors and their consequences have astonished, during the last years, both healthcare professionals and ordinary people. As a matter of fact, both scientific literature and mass media are more and more sensitive to the problem. But to tackle the problem, we must go from the level of statistics and generic complaint to the individual level, to understand how and why errors happen. This is not for blaming an individual, but for reconstructing the pattern of actions that led up to the error. Opposite to the first impression, often this procedure relieves the individual, by discovering a chain of responsibilities. It is agreed that "changing the culture of blame requires a revolution", and we do not pretend that Information and Communication Technology (ICT) makes the miracle. However, without ICT the miracle is even more difficult. In my talk I will show how ICT applications, and in particular Business Process Modelling, can be used to better manage medical processes, to discover bottlenecks, pitfalls and errors and to reason about the causes and consequences of them. This is an opportunity for healthcare organisations to learn from their errors and to spread this new culture within other healthcare communities. As long as a non-compliance with a protocol or clinical practice guideline can be considered an error (that's not necessarily true), I will describe how protocols and guidelines can be implemented exploiting workflow technology, and how compliance with them can be used as an indicator of healthcare delivery quality. Moreover, I will describe an example of ICT implementation that follows the so-called "socio-technical approach", that makes the individual aware to contribute to the progress of a community of practice. I will also illustrate process mining results from a stroke registry data: different processes have been mined in different hospitals, referring to the same type of patients. Physicians' reaction when looking at the detected differences will be reported. Eventually, the challenges in applying process mining techniques in healthcare, with respect to other settings, like the industrial or commercial setting, will be discussed.

A concept for the assessment of electronic communication in integrated information systems

Samrend Saboor[1], Elske Ammenwerth[1]

[1] Institute for Health Information Systems, University for Health Sciences, Medical Informatics and Technology (UMIT), Eduard Wallnöfer-Zentrum 1, 6060 Hall in Tyrol, Austria
{Samrend.Saboor, Elske.Ammenwerth}@umit.at

Abstract. Integrated hospital information systems are complex architectures that are grown over the course of years. Changing such architectures is challenging – negative side effects are hardly predictable. The objective of this paper is to present a concept that adequately describes communication processes and systematically detects potential errors. Qualitative content analysis and problem-centered expert-interviews were applied to collect communication errors and their reasons in a structured categorization. The categorization was used to derive requirements for the description of communication processes and the detection of errors. The developed concept describes communication processes as chains of application systems that transfer instances of information objects via communication interfaces. The concept was developed considering common problems of the electronic communication in integrated information systems. The evaluation of the concept is planned. The resulting concept could assist hospital information management to plan changes for the information system and to foresee severe conflicts.

Keywords. Hospital information management, System architecture, Classification

1 Introduction

In recent years, the electronic acquisition and transmission of information objects, e.g., order forms or examination reports, gained importance in the health care domain [1]. The correct transmission of information objects between the involved computer-based application systems (e.g., order entry system) has become vital for processes in health care institutions [2]. In this context, terms such as “seamless integration” are used quite commonly (e.g., [3, 4]) and underline the necessity that all communication partners must share conventions that enable them to effectively operate together [5]. These conventions pertain to the structure of information objects (e.g., as sets of identifier-attributes pairs), the meaning of each attribute and consequently each

application system's communication interfaces. In the health care domain, the two most important communication standards are DICOM (Digital Imaging and Communications in Medicine, [6]) and HL7 (Health Level 7, [7]).

Nevertheless, communication within heterogeneous information systems is still error-prone [8, 9] . Even the use of just one of the standards requires additional implementation efforts [10, 11]. Among others, this situation is caused because standard definitions still allow the misuse of objects and services. Therefore, the international initiative IHE (Integrating the Healthcare Enterprise) provides a framework for the coordinated use of established standards [12, 13]. In accordance to the process-oriented paradigm that has been increasingly propagated for the healthcare domain in the last years, the IHE defines transactions in its technical framework [14]. These transactions are organized in distinct integration profiles that represent all common use cases. Each profile defines those transactions wherein the application systems are regarded as actors that exchange specific information objects while acting in a certain way.

The IHE helped to improve the integration within heterogeneous health care information systems. This is important because fully interoperable communication interfaces between all involved application systems are crucial prerequisites for correct information logistics (i.e., the right information at the right time and place in the right form to the right people, so that these can decide correctly (e.g., [15])) or rather process-oriented information systems (i.e., the deployment of Workflow Management Systems, e.g., [16]). However, correct information exchange on a broad scale still cannot be taken for granted [17, 18]. Communications that appear successful can still have mistakes in the exchanged content while the underlying reasons are difficult to identify. Even changes in properly working infrastructures may cause bad side-effects that are hardly predictable (e.g. [13]) – also having in mind that process-oriented information systems must adapt any organizational changes. Problems related to communication infrastructure and processes must be examined carefully because they affect the quality of patient treatment. It is, therefore, important to detect any possible communication errors. However, the current methods seem to not be optimal for the assessment of computer-based communication processes [19] – they either perform e.g. time measurements (e.g., MOSAIK-M [20]) or reachability analyses on Petri-net based models in order to detect bottlenecks and the best performing variations [21]. Others, such as the method of Alexopoulos et al. [22], were developed to analyze those processes that show significant variations in their executions. While some of these methods do not include information objects, others provide them in a simple way per default (i.e., information objects are either simply represented as named objects that are included in the process models without further adjustments or they are regarded as containers for arbitrary collections of attributes that are not filled per default). In the latter case, information objects are not defined on a formal base and thus cannot be utilized by assessment methods that concentrated on the detection of communication processes.

Therefore, a concept is needed that considers the characteristics of information objects and their processing in heterogeneous information systems – which would also facilitate the deployment of process-oriented information systems. The objective of the present paper is to present a concept for describing communication processes and the systematic detection of potential errors in these processes.

2 Methods

An essential part of the new approach is an adequate description of the communication process. Here, "adequacy" means that all the details that are necessary to detect potential communication conflicts are included in the model. Therefore, it is important to start with a collection of common communication problems as well as the conditions under which these problems occur. The latter are also required for the second aspect of the new concept – the detection of potential communication conflicts. Based on the collected information, queries for each of the communication problems are developed, which can be applied to the modeled processes.

Therefore, it makes sense to start with the collection of the communication problems and their prerequisites. The results of this first step must be collected in a categorization. This is required in order to efficiently examine the collected results and to derive the requirements for the description notation in the second step. The whole process consists of three steps that are explained in more detail in the following sections:

1. Collection of communication problems and their prerequisites
2. Validation of collected results
3. Deduction of requirements for the modeling notation and model assessment

2.1 Collection of communication problems and their prerequisites

In order to collect communication conflicts and their prerequisites, we chose the method of subsuming qualitative content analysis (according to Mayring [23]) – this type of content analysis aims to filter the main content by the abstraction and dynamic declaration of categories. For this we conducted a systematic review of the available literature in PubMed. Here, we made use of the experiences from earlier projects in the area of process assessment (e.g., [24]). The inductive approach that we chose is shown in Figure 1:

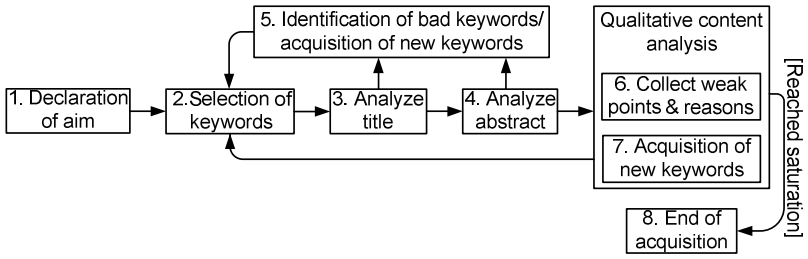

Fig. 1. Process of literature review – After declaring the review's aim, appropriate keywords and references are searched iteratively. Adequate results are analyzed for communication weak points and their respective reasons. The process stops after an observable saturation.

First, an initial set of search phrases (e.g., "quality information processing") was declared according to the aim of the review. In steps three and four, the title and

abstract of all the references that resulted from the search phrases were reviewed. In step five, we adjusted and augmented our search phrases according to the adequateness of the resulting references. In this way we found 4,188 references. These references dealt with, among others, information management (e.g., [25]), reports on integration projects in the field of Hospital Information System (HIS), Radiology Information System (RIS) and Picture Archiving and Communications Systems (PACS) (e.g.,[26]). From this set we dropped all of those references that were older than 20 years, which dealt with the implementation of very specialized software applications or dealt with organizational issues. In the sixth step, we performed the actual qualitative content analyses on the remaining 426 references. In this step, further keywords were also found, in which we used in step seven to adjust our set of search phrases. The process stops when the saturation of new errors is reached.

Relevant excerpts from the references were first collected in an unsorted list. During transcription, the formulations of similar entries were assimilated and grouped. In this way a categorization of communication problems and their prerequisites was created. Table 1 in section 3 shows parts of this categorization.

2.2 Validation of communication problems and their prerequisites

In order to ensure the validity and reliability of the collected communication problems and their prerequisites, we chose the method of problem-oriented expert interviews. For this purpose, we interviewed experienced hospital information managers, integration engineers (i.e., software engineers specialized on the implementation of medical communication standards such as HL7 and DICOM) and members of working groups dealing with integration issues. The exclusive criteria for the selection of the experts were:

1. At least two years' (preferably five years) working experience in implementing clinical application systems that communicate via HL7 or DICOM.
2. At least two years' (preferably five years) working experience in administrating HIS, PACS or RIS databases (i.e., database is filled with data from HL7 and/or DICOM messages)
3. At least two years' (preferably five years) practical experience in projects introducing or updating clinical application systems that communicate via HL7 or DICOM.
4. At least two years' (preferably five years) practical experience in implementing IHE profiles.

In total 42 experts were invited to participate. 28 experts responded and were willing to review the categorization. In total, 16 experts were actually interviewed. They received an electronic copy of the error categorization and basic review guidelines that clarified the focus of the interview. The actual interview was mostly performed via telephone as an open semi-structured interview.

2.3 Deduction of the requirements for the modeling notation and model assessment

After completing the expert interviews, the details in the categorization were used to develop queries for the detection of each problem and also to derive the requirements for the modeling notation. Here, the basic idea was to develop clear expressions in terms of propositional logic. Each of these expressions shall concisely describe the conditions for its respective problem:

Thus, we started by extracting the main atomic statements from the collected reasons and requirements for each of the problems. For example, the requirement "*Required data, e.g., for creation of new information objects, shall be retrieved automatically from sources with assured data quality*" contains the two atomic statements "V1: *NOT(data provided via electronic interfaces)*" and "V2: *NOT(data provided from source of assured quality, e.g. db-content)*" (Note: recommendations were negated in order to describe conditions for a negative event – this was done by applying negation according to the laws of De Morgan). This resulted in a redundance-free repository of atomic statements with each identified uniquely.

In the next step, the atomic statements were used to rebuild the original reason- and recommendation-entries in a uniform way. For example, the recommendation mentioned above would be then expressed by the logical term: "V1 AND V2".

An additional aggregation of the logical terms was necessary because some of the collected communication problems had multiple reasons and recommendations. We therefore combined the logical terms according to the disjunctive normal form and summarized them by applying Karnaugh maps.

The atomic statements mentioned above determined which details have to be acquired in order to be able to receive a result out of those logical terms. For instance, the atomic statement V2 requires to determine whether the used data source is recognized as a data base of good information quality – this could be expressed and later on considered as a boolean attribute (i.e., true or false).

3 Results: A concept for the detection of communication errors

The results are presented according to the steps that are described in the methods section: The qualitative content analysis and the subsequent experts interviews resulted in a classification of communication problems and their prerequisites. After presenting parts of this categorization in section 3.1, the developed concept is explained in section 3.2.

3.1 Categorization of communication problems

The resulting categorization consists of five hierarchy levels. Table 1 shows an excerpt of the categorization – further details can be found in [27]:

- The top level (i.e., "Aspect" in Table 1) distinguishes whether the problems are related to single information objects or rather series of these, the administration of

information objects or the transfer of information objects between application systems.

- The second level (i.e., "Detailed aspect" in Table 1) substantiates the separation into more concrete aspects such as "Content" or "Acquisition and import". This level comprises 10 entries.
- The third level (i.e., "Problem class" in Table 1) groups similar concrete problems. The entry "Content" on the second level contains e.g. the problem classes "Wrong details in data" and "Missing data". The categorization contains 29 of these classes.
- The fourth level (i.e., "Problem" in Table 1) contains the actual results of the qualitative content analysis. The categorization has 81 problem entries.
- The fifth level contains reasons for the errors on level four along with recommendations that the authors gave in order to avoid those problems (i.e., "Reason" and "Recommendation" in Table 1). This level comprises 229 entries.

Note: The columns "Reason" and "Recommendations" are independent although they are both assigned to the related entry in column "Problem".

Table 1. Excerpt of the categorization of communication errors (i.e., column "Problem") and their prerequisites (i.e., columns "Reasons" and "Recommendations")

Aspect			
Detailed Aspect			
	Problem Class		
	Problem	**Reason**	**Recommendations**
I) Information objects/Series of information objects *(i.e., errors that are related to single information objects or series of these)*			
I.1) Content *(errors that are related to content problems)*			
	I.1.a) Wrong details in the data *(errors dealing with erroneous content)*		
	Data entry error/Editing error	Manual data entry	requires[Automated checks of manual data entry]
		Too many entries in the worklist/manual selection (ambiguous)	requires[Required data, e.g., for creation of new information objects, shall be retrieved automatically from sources with assured data quality]
		Missing standardization of entry forms allows typos during manual entry	requires[Usage of standardized tags/attributes instead of proprietary ones]
		Combination of different independent details in one data field	requires[Usage of automatically provided content, e.g. worklists, instead of manual data entry]
		Incompatible/foreign char-sets	requires[appropriate input-interfaces e.g. drop-down boxes rather than typing] - e.g., replacement of accented letters with unaccented letters, conversion of all strings to upper/lower-case, replace punctuation signs with a space, discard non-informative spaces
		Transformation of the original content	
	Redundant data entry	-	requires[Support of modality worklist]
	Wrong identification of	Wrong labeling by staff	requires[Unambiguous identification of single information objects and collections of those]

information objects		requires[Generation/Provision of new identifiers after manipulating the content of information object]
Wrong details in the data/Inaccurate details in the information objects/ Corruption of content	Conversion of an information object into another standard	requires[Patient data are only imported into and managed in the HIS and are distributed from there]
	Inconsistent usage of enumerable values (e.g., country)	requires[A central system for managing patient- & order-data]
	Different vendors implement standardized services in a proprietary way (e.g., when information objects are first stored into a database and then new information objects are created out of the database fields)	requires[fixed format for enumerable values]
		requires[Purely digital/electronic interface allows for the automated transmission of demographic data]
I.1.b) Missing data *(errors dealing with incomplete information objects or incomplete series of information objects)*		
Missing identification of information object instances	Incompatible identification attributes between requester and provider	requires[unambiguous identification of each information object instance] or rather: leaking availability (because information objects was lost)/wrong assignment of information object to patient
	Identifying attributes differ between standards/sets of identifiers are not coherent between different standards	requires[Each instance of an information object must be identified uniquely]

3.2 Concept for the description and assessment of communication processes

The details of the categorization were used to derive requirements for the adequate description of communication processes. Based on these requirements, a concept for the description of computer-based communication and the detection of potential conflicts was developed. In the following, some of these requirements are listed:

- The process description must include whether the information objects are filled/edited via manual or automatic data entry. This is important because manual data entry can cause errors in the content of information objects (see Table 1 entry I.1.a – Data entry error/Editing error).
- The process description must clarify which attributes are used for identifying the information objects. This is important because missing or wrong identification numbers can cause the loss of information objects (see Table 1 I.1.b – Missing identification of information object instances).
- The process description must clarify to which communication standard each communication interface of the application systems are dedicated to and specifically to which service they are assigned to. This is important because incompatibilities can cause the corruption or loss of information objects.
- The process description must clarify the character set of the content in information objects. This is important because the content can be corrupted when the sending and receiving application systems use different character sets.

– The process description must clarify whether the creation/editing of an information object instance is completed. Therefore, the processing state must be represented. This is important because otherwise the information objects are either not completed or finished information objects (e.g., signed documents) and are edited again, which should not happen.

These and further derived requirements were used to develop a concept in which communication processes consist of two types of elements: application systems and information objects (see Figure 2). The processes are built as chains of application systems that transfer instances of information objects via communication interfaces.

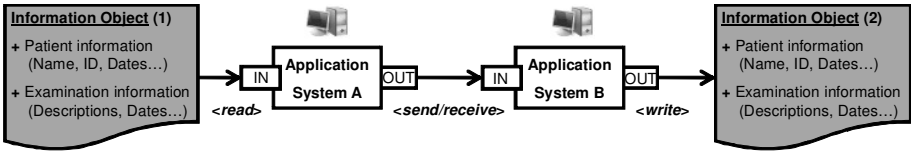

Fig. 2. Communication processes are described by chains of application systems that transfer information systems via communication interfaces.

Figure 3 shows a more detailed view on the interface design – it shows how the requirements that were mentioned at the beginning of this section were considered in the developed concept: Each interface is dedicated to a specific communication convention or rather standard (e.g., HL7, DICOM or proprietary) and, as far as possible, one of its services. These details are contained in the "Administrative" part of each interface. The "Content" part describes the location, value representation, value length of each message attribute and whether it is required or not. In the case of output-interfaces, the resulting information objects/messages are written according to the interface specification. In the case of input-interfaces, the received information objects are read according to the convention as defined in the interface specification.

Further, Figure 3 also shows an example of how the concept could help to detect potentially missing attributes. In the depicted case (marked by a circle), the application system B expects an order ID and thus declares this attribute as "required"– application system A, however, declares this attribute as optional and does not provide a value. The association between the related fields of the different interfaces could be, for instance, implemented using label-based scheme matching – using the Idx-values to identify equivalent interface fields. Such a scheme matching is presumed by the example in Figure 3 (Note: for exemplary purposes, e.g., Figure 3, the matching was carried out manually).

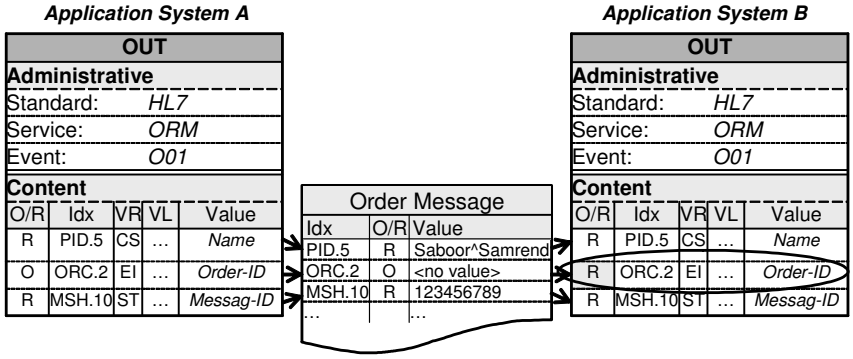

Fig. 3. Detailed view: The output-interface of application system A sends an order request message via HL7 ORM to the input-interface of application system B. Here, "O/R" means optional or required attribute, "Idx" is the index of the attribute within the sent message, "VL" is the value length and "Value" is the actual content.

4 Discussion

Process-oriented information systems rely on properly integrated application systems. In integrated hospital information systems the electronic communications are complex interactions of application systems. Difficulties arise due to partly or completely different interpretations of common communication standards. These difficulties can corrupt the content of the transmitted information objects – leading to the loss or wrong assignment of these objects. Consequently, the related patient treatment process is negatively affected. The discussion first provides a small overview over the currently established process modeling notations. After this, the strengths and weaknesses, or rather open issues of the newly developed concept, are discussed.

4.1 Overview of the established process modeling methods

The following overview focuses on those aspects that are related to the communication of information objects. Each of the introduced methods is discussed according to its capabilities of modeling the processing and communication of information objects:

- Activity Diagrams of the Unified Modeling Language (UML) [28]:
  Among the different notations of the UML the Activity Diagrams are the most suitable for describing the control flow of working processes. It is also possible to assign instances of generic classes to each of the process actions as inputs or outputs. The effect can in turn affect the instances that then change their state. Thus, the Activity Diagrams – seen as a generically applicable method – could be used to also model the communication processes. This would, however, require additional extensions e.g. the definition of some kind of information model.

- Event-driven Process Chains (EPC) of the ARIS Toolset [29]:
  The EPC are a part of the ARIS Toolset that comprises different views on organizations through specialized sub-models. The EPC belong to the most established notations for modeling business processes. As such, they focus on the description of control flows but also contain elements for generic documents and information objects. It is also possible to detail these objects. However, modeling information processing on a technical layer (i.e., considering details that deal with e.g. communication standards, data conversion or the location and identification of attributes) is not in the primary scope of the EPC.
- MLDesigner:
  The MLDesigner method originates from the domain of system- and control-theory and is, therefore, more technically focused. The processes are modeled in terms of modules that are combined to block diagrams. Each module has input- and output-signals and its behavior can be implemented using the C++ programming language. Although there are libraries of pre-defined modules for different purposes, the description of integrated communication processes is not possible without further adjustments.
- Modeling, simulation and animation of information- and communication-systems in medicine (Mosaik-M):
  MOSAIK-M models are used for planning and analyzing existing or new information systems within business process re-engineering projects. These models are instances of an own meta-model that combines actors (i.e., computer system or human actor) and functional units. Software scripts are used to describe each actor's behavior. It is possible to model information objects on a level as detailed as in our concept if this is necessary for the related re-engineering project. Because of its openness to extensions, the MOSAIK-M could be adjusted/extended using our concept as a guideline.

We conclude that all the regarded methods need further adjustments in order to describe communication processes in a way that they could be used for our purposes (i.e., modeling electronic communications and systematically detect possible errors).

4.2 Strengths and open issues of the new concept

Experiences from earlier projects (e.g., [30]) show that the inclusion of human factors, e.g., staff overview, communication between human actors as well as the assessment of their conversation quality, are hardly assessable in an automated way. Thus, the newly developed concept concentrates on the description of computer-based communication. It considers issues related to the application of the established communication standards in the domain of medical informatics (i.e., DICOM and HL7). Specifically, it was designed considering the known problems that occur in projects dealing with the integration of new application systems into information systems etc. A systematic review of such experience reports and interviews with domain experts shall guarantee that no severe problem is omitted. The problem reasons that were also acquired are important to detect potential conflicts in even complex models. However, the concept has to be evaluated using complex communication processes from a real hospital environment. This particularly pertains

the concept's assessment aspect – it must be examined whether the checking routines effectively allow for the detection of specific communication problems. Moreover, a software tool is needed to facilitate the modeling work. It would be possible to use the concept as a guideline for adjusting the software tools of established methods such as e.g. MOSAIK-M or MLDesigner. This aspect has to be examined separately.

5 Conclusion

Process-oriented information systems require that all application systems that are included in workflow descriptions can properly communicate with each other. However, integrated hospital information systems are complex architectures that are historically grown. Upgrading or changing the whole system or just parts of it is a challenging task – negative side effects of these changes are hardly predictable. We, therefore, developed a concept for the adequate description of electronic communication processes and the systematic detection of possible errors in these processes. For this purpose, the developed concept considers the known problems and their respective reasons that are collected together in a structured categorization. The developed concept describes communication processes as chains of application systems that transfer instances of information objects via communication interfaces. It is planned to evaluate the new concept using real communication processes of sufficient complexity. The resulting concept could assist hospital information management to plan changes to the information system and to foresee severe conflicts.

6 References

1. Haux R. Health information systems - past, present, future. Int J Med Inform 2006;75(3-4):268-81.
2. Coiera Enrico. When conversation is better than computation. JAMIA 2000;7(3):277-286.
3. Brazhnik O., Jones J. F. Anatomy of data integration. J Biomed Inform 2007;40(3):252-69.
4. Giuse D. A., Kuhn K. A. Health information systems challenges: the Heidelberg conference and the future. Int J Med Inform 2003;69(2-3):105-14.
5. Kun L. Interoperability: the cure for what ails us. IEEE Eng Med Biol Mag 2007;26(1):87-90.
6. ACR/NEMA. DICOM Homepage. 2007 [cited 2007 Jun]; Available from: http://medical.nema.org
7. HL7. Health Level 7. 2007 [cited 2007 Jun]; Available from: http://hl7.org
8. Hammond K. W., Helbig S. T., Benson C. C., Brathwaite-Sketoe B. M. Are electronic medical records trustworthy? Observations on copying, pasting and duplication. AMIA Annu Symp Proc 2003:269-73.
9. Khorasani R. Medical management: expanding radiologists' role using information technology to improve the quality of care. Semin Roentgenol 2003;38(3):282-6.
10. Gross-Fengels W., Miedeck C., Siemens P., Appel R., Muckner K., Finsterbusch J., et al. [PACS: from project to reality. Report of experiences on full digitalisation of the radiology department of a major hospital]. Radiologe 2002;42(2):119-24.

11. Matthews J. W., Bosch W. R. Explicit-VR transfer syntax limits the value multiplicity of DICOM data elements with decimal string (DS) value representation. Phys Med Biol 2006;51(5):L11-2.
12. HIMMS/RSNA. IHE - changing the way healthcare connects. 2007 [cited 2007 Jun]; Available from: http://www.ihe.net
13. Mildenberger P., Wein B., Bursig H. P., Eichelberg M. [Current developments in DICOM and IHE]. Radiologe 2005;45(8):682-9.
14. Reichert M, Dadam P. Towards Towards Process-oriented Hospital Information Systems: Some Insights into Requirements, Technical Challenges and Possible Solutions. In: Proceedings of the 43th GMDS (GMDS '98), Bremenm 1998, pp. 175–180.
15. Augustin S. Information als Wettbewerbsfaktor: Informationslogistik - Herausforderung an das Management. Köln; 1990.
16. Dadam P, Reichert M, Kuhn K. Clinical Workflows – The Killer Application for Process-oriented Information Systems. Proceedings of the 4th Int. Conference on Business Information systems, Poznan, Poland, April 2000, pp. 36-59.
17. Boochever S. S. HIS/RIS/PACS integration: getting to the gold standard. Radiol Manage 2004;26(3):16-24; pp. 25-7.
18. Carr C. D., Moore S. M. IHE: a model for driving adoption of standards. Comput Med Imaging Graph 2003;27(2-3):137-46.
19. Brigl B, Strübing A, Wendt T, Winter A. Modeling interdependencies between information processes and communication paths in hospitals. Methods Inf Med 2006;45(2):216-24.
20. Hoffmann I, Bergmann J, Bott O.J, Pretschner D.P. Einsatz einer rechnergestützten Modellierungs- und Simulationsumgebung für den Entwurf telemedizinischer Systeme am Beispiel von MOSAIK-M. In: Steyer G, Tolxdorff T, editors. TELEMED; 2005 2005; Berlin: Aka-Verlag; 2005. p. 309-20.
21. Blake J., Carter M., O'Brien-Pallas L., McGillis-Hall L. A surgical process management tool. Medinfo 1995;8 Pt 1:527-31.
22. Alexopoulos C, Goldsman D, Fontanesi J, Sawyer M, De Guire M, Kopald D, et al. Healthcare I: a discrete-event simulation application for clinics serving the poor. In: Winter Simulation Conference - Proceedings of the 33nd conference on Winter simulation; 2001. p. 1386-91.
23. Mayring Philipp. Einführung in die qualitative Sozialforschung: eine Anleitung zu qualitativem Denken. 5., überarb. Aufl. ed. Weinheim: Beltz, Psychologie-Verl.-Union; 2002.
24. Saboor S, Chimiak-Opoka J, Ammenwerth E. Supporting the systematic assessment of clinical processes: the MedFlow method. Methods Inf Med 2007;46(5):586-594.
25. Haux R, Winter A, Ammenwerth E, Brigl B. Strategic Information Management in Hospitals. New York, USA: Springer-Verlag; 2004.
26. Kotter E., Langer M. Integrating HIS-RIS-PACS: the Freiburg experience. Eur Radiol 1998;8(9):1707-18.
27. Saboor S, Ammenwerth E. Assessing communication processes within integrated health information systems. In: Proceedings of the Sixth IASTED International Conference on Biomedical Engineering; 2008; Innsbruck, Austria; 2008.
28. Object Management Group. Object Managment Group - UML. 2008 [cited 2008 May 14]; Available from: http://www.uml.org
29. AG IDS Scheer. IDS Scheer AG - Country Site DE: ARIS Software (Modeling): Ereignisgesteuerte Prozessketten. 2008 [cited 2008 May 14]; Available from: http://www.aris.de/de/Software/EPK_Modellierungsstandards/79890.html?mod_srch[result_link]=1
30. Saboor S., Chimiak-Opoka J., Ammenwerth E. Supporting the Systematic Assessment of Clinical Processes: the MedFlow Method. Methods Inf Med 2007;46(5):586-94.

Petri Nets as a formalism for comparing expressiveness of workflow-based Clinical Guideline Languages

María Adela Grando[1], David W. Glasspool[1], and John Fox[2]

[1] School of Informatics, University of Edinburgh,Edinburgh, UK,
{dglasspo,mgrando}@inf.ed.ac.uk
[2] Department of Engineering Science, University of Oxford, Oxford, UK
john.fox@eng.ox.ac.uk

Abstract. There has been relatively little work on formal analysis of expressiveness and verification of structural, behavioural and temporal properties in clinical workflow. In this paper we discuss Coloured Petri nets (CPNs) as a formalism to support such analysis. We show in detail how a typical clinical guideline language (PRO*forma*) may be formally mapped to a CPN representation, then show how such a mapping allows formal proofs that a guideline language is capable or is not capable of expressing a standardised workflow pattern.
Keywords: clinical process modelling, care pathways, petri nets, workflow patterns

1 Introduction

A number of process-based languages for the specification of medical guidelines have been developed by the Health Informatics community, allowing care plans, care pathways and protocols to be expressed in well-defined, computer-interpretable formal languages. As well as being easier to disseminate and maintain than traditional paper-based guidelines computer-interpretable guidelines (CIGs) have the potential to make a significant contribution to quality and safety in healthcare while also reducing costs [5] [4].

The formalisation of an informal, text-based clinical guideline in a well-defined language also represents an essential first step towards a more ambitious goal: the formal analysis of clinical guidelines [2] [14]. This offers the possibility of a formal strategy for analyzing the gap between the CIG and the original paper guideline, to check for anomalies like ambiguity, incompleteness, inconsistency, violation of regulations and constraints etc. and to formally compare medical guidelines expressed in different CIG languages or to compare the expressive capabilities of the languages themselves.

The next step towards this goal is to map guidelines expressed in existing CIG languages into a standard form that allows formal workflow analysis techniques to be brought to bear. In this role we are particularly interested in approaches that would allow a formal foundation for process descriptions. Two stand out:

1. *Process algebra.* This has been used to provide workflows with formal semantics [15], to prove equivalence of process specifications and to verify workflow structural and behavioural properties [12]. Pi-calculus is a popular process algebra variant that provides the notion of mobility for modeling and formal analysis of workflows [9]
2. *Petri Nets (PNs).* This approach can be used to study expressiveness of formal process languages and to verify structural, behavioural and temporal properties of process definitions.

While PNs are based on (bipartite) graphs, process algebras are based on a textual description. Many notions developed for Petri nets have been translated to process algebra, and vice versa. However PNs and their higher level variants (CPNs, time PNs and hierarchical PNs) are growing in popularity as a formal process representation framework for workflow [3] [16] [1]. PNs have a strong mathematical and formal foundation, they have well-known and standardised formal semantics, they have a standard and intuitive graphical representation and they are vendor independent. PNs have also been proved to be adequate for specifying the primitives needed to represent workflow processes [16]. Standard algorithms exist for the verification of structural, behavioural and temporal properties and for simulation of processes expressed in PNs. (E.g. In [10] medical guidelines are mapped into PN form for validation, simulation and optimisation of resource allocation). The expressiveness of PNs can be studied and compared using the notions of simulation and equivalence bisimulation [6].

For these reasons we are exploring the use of PNs as a standard formalism for formal analysis of clinical workflow. In this paper we consider the topic of expressiveness. We show how a CPN approach may be used to formally prove that a CIG language is capable of expressing a standardised workflow pattern, and (possibly more importantly) to prove that a CIG language is *not* capable of expressing a particular workflow pattern.

Although the principles are intended to be general, for concreteness we base our examples on our own CIG language, PRO*forma*. We therefore begin by presenting a mapping from PRO*forma* task representations into CPNs, before using this mapping in two examples demonstrating the two types of proof.

2 Mapping PRO*forma* to the Petri Net formalism

2.1 PRO*forma*

PRO*forma* is an executable process modelling language that has been successfully used to build and deploy a range of decision support systems, guidelines and other clinical applications. It has a declarative syntax and a well-defined operational semantics [13]. A PRO*forma* guideline consists of a small set of task classes that can be composed into networks representing arbitrarily complex plans or procedures, and a similarly small set of attributes which control task enactment. PRO*forma* bases its process model on a minimal ontology of

tasks, the main task classes are actions, enquiries, decisions and plans. Actions represent a procedure to be executed on the external environment (e.g. administering a drug or updating a database). Enquiries are tasks carried out to acquire information from some person or external system. Decisions are processes for making choices about what to believe or what to do. Plans are collections of tasks grouped together for some reason, perhaps because they share a common goal or use a common resource, or need to be done in a synchronized way. All four task types share common attributes inherited from the root class in the task ontology (called a keystone).

Tasks have a small set of generic attributes, including:

- *State:* which can take the values *Dormant*, *InProgress*, *Discarded* and *Completed*. A task that has not been started is *Dormant*; a task is *InProgress* if it has been started but not yet *Completed* or *Discarded*; it is *Discarded* if the logic of the guideline implies either that it should not be started or that it cannot be completed, and *Completed* if it has been carried out successfully. Tasks may be cyclic, that is to say that they may be executed repeatedly during enactment of a plan. Transition from *Completed* to *InProgress* occurs when the task itself cycles and transitions from *Completed* and *Discarded* to *Dormant* occur when its parent plan cycles.
- *Antecedent_tasks:* a sequence of task identifiers that indicates the tasks that must be completed before this one starts.
- *Precondition:* a condition that must be satisfied for the task to start execution. If the precondition is not satisfied and the task has no other constraints associated, then the task becomes *Discarded*.
- *Postcondition:* a condition that is assumed to be true when the task completes.
- *Trigger (event trigger):* an externally introduced message allowing tasks to be explicitly started without waiting for their scheduling to be satisfied. During the enactment of a plan an external message can be triggered multiple times and for each time it is introduced the associated tasks get active.
- *Wait_condition (state trigger) :* if a state trigger becomes true then the associated task will be immediately considered for enactment if its preconditions are satisfied.

Particular subclasses of tasks have distinctive properties, for example plans have:

- *Termination_condition:* a sufficient (though not necessary) condition for successfully terminating the current plan enactment, changing its state to *Completed*.
- *Abort_condition:* a sufficient condition to abort the plan and cancel downstream tasks.

Four conditions, *Start(X), Discarded(X), Cycle(X)*, and *Completed(X)* may be defined [13] to determine unambiguously at run time the transitions between states of a task *X*. *Start(X)* is true iff the parent plan of *X* (if any) is *InProgress* and either: its scheduling constraint conditions are true and it has no trigger;

or it has been triggered; or it is a cyclic task which has completed and needs to begin a new cycle. *Discarded(X)* is true iff any of the following are true:

1. Task X is *Dormant*, *InProgress*, or is a cyclic task about to be repeated, and its parent plan has been *Discarded* or *Completed*, or
2. Task X is *Dormant*, its parent plan is *InProgress*, its scheduling conditions are true, and either it has antecedent tasks that have all been discarded, or it has a precondition that is not true, or
3. Task X is a plan and its abort conditions are true.

Completed(X) is true if task X is currently *InProgress* and (if not a plan) it has been successfully executed, or (if it is a plan) all of its component tasks are completed or discarded. *Cycle(X)* is is a condition that must be satisfied for a cyclic task to be repeated. It may specify either a number of cycles to be completed or a condition that must become true for cycling to stop.

The condition *Start(X)* is checked first. Only if this is not satisfied will the *Discarded(X)* condition be checked, and only if the *Discarded(X)* condition is not satisfied will the *Completed(X)* condition be checked. In the left part of figure 2 we present the finite state transition system for PRO*forma* tasks.

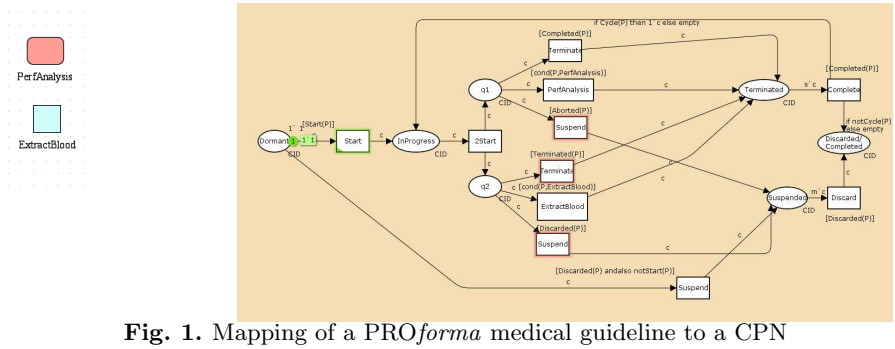

Fig. 1. Mapping of a PRO*forma* medical guideline to a CPN

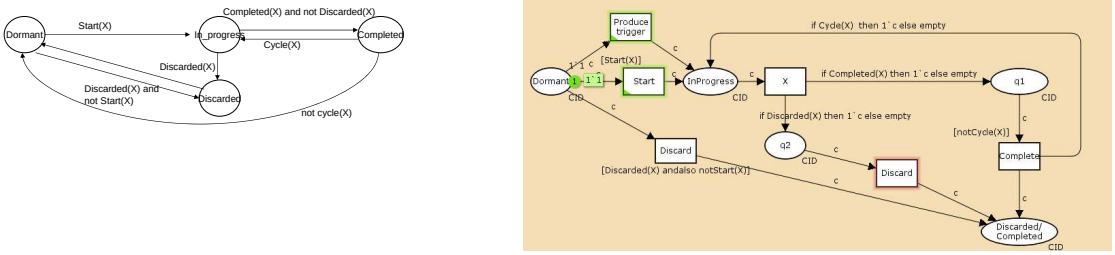

Fig. 2. Mapping of a PRO*forma* task to a CPN

PRO*forma* processes may be represented diagrammatically as directed graphs in which nodes represent tasks and arcs represent scheduling constraints. By convention plans are represented as rounded rectangles, decisions as circles, enquiries as diamonds and actions as squares, with the generic task shown as a keystone shape. A standalone application, such as a guideline or care pathway, typically consists of a single plan which may be recursively divided into sub plans. In the left part of figure 1 we present a PRO*forma* plan called *MedicalGuideline*. This is composed of two parallel tasks: *PerfAnalysis* (a plan corresponding to perform analysis) and *ExtractBlood*(an action).

2.2 Coloured Petri Nets

Coloured PNs are high-level PNs where each token has associated a type called token colour. Coloured PNs consist of *places*, *transitions*, and conditional directed *arcs*. Graphically places are represented with circles, transitions are drawn as boxes and arcs are represented with arrows. Arcs connect places with transitions and transitions with places. When arcs connect places with transitions they can have a condition associated. Places may contain any number of tokens of a certain colour. For a transition to be able to occur there must be sufficient coloured tokens that match its condition. Then a binding between the tokens chosen and the variables used in the condition of transition takes place. When a transition is enabled the multi-set of tokens that were bound due to incoming arcs are removed from the corresponding input-places and a multi-set of tokens is added to each output-place. Coloured PNs are used in [11] to specify patterns because they can simulate nondeterministic and concurrent behaviours. From a set of $m \geq 0$ enabled transitions, n of them can be non deterministically chosen to be simultaneously executed, $n \leq m$. Besides for a given transition multiple bindings can occur concurrently iff there are enough coloured tokens for the chosen bindings.

2.3 Mapping PRO*forma* tasks to Coloured Petri Nets

In this section we define two algorithms to map from PRO*forma* to CPNs. An arbitrary PRO*forma* task X of type decision, enquiry or action can be mapped into a CPN by means of algorithm 1. An arbitrary PRO*forma* plan P can be mapped into a CPN by means of algorithm 2, where for each task in P of type plan algorithm 2 can be recursively applied.

To allow the composition of plans the CPN $PROfCPN$ resulting from applying algorithm 2 to a plan P has a unique starting place called *Dormant* and a unique final place called *Discarded/Completed* that can correspond to termination due to discarding or completion of plan P. In PRO*forma* concurrent execution of tasks is possible, therefore a transition *nStart* is added to $PROfCPN$ such that after a token is generated in the *Dormant* place it is possible to fire the transition *nStart* and generate one token for each one of the n potential initial tasks in P.

In PRO*forma* once the completion or discarding condition of a plan P is satisfied all the tasks that are pending for execution are canceled. Therefore for each transition T_X in $PROfCPN$, corresponding to a non terminal task X in P, we define $t+2$ subsequent transitions: t transitions $T_{Y1}, ..., T_{Yt}$ corresponding to subsequent tasks $Y1, ..., Yt$ of X, the transition $Terminate$ and the transition $Suspend$. If T_X corresponds to a terminal task X in P, then we define only $Terminate$ as the subsequent transition of T_X. In this way in case the completion (discarding) condition of P is satisfied all the s (m) transitions pendent of execution in $PROfCPN$ perform the $Terminate$ $(Suspend)$ transition reaching state $Terminated$ $(Suspended)$. From state $Terminated$ $(Suspended)$ the s (m) flows of execution become one token after the execution of transition $Complete$ $(Discard)$. In case that after performing the transition $Complete$ more cyclic executions of plan P need to be enact then one token flows to state $InProgress$. If the cyclic condition of plan P is not satisfied then after the execution of transition $Complete$ one token flows to the state $Discarded/Completed$. If the transition $Discard$ is executing then one token is fired to place $Discarded/Completed$.

The flow of execution that takes place when a PRO*forma* plan is discarded or completed resembles the one described by pattern WCP-20 (cancelation case) from [11]. The only difference is that in PRO*forma* the discarded or completed state is not reached due to the introduction of a trigger, but because of the satisfaction of predicates *Discarded(P)* or *Completed(P)*. Therefore knowing pattern WCP-20 was be very advantageous to specify the mapping of a PRO*forma* plan into a CPN.

1. Algorithm to map PRO*forma* tasks of type decision, enquiry or action to Petri Nets

An arbitrary PRO*forma* task X of type decision, enquiry or action can be mapped into the CPN presented in the right part of figure 1, where:

- Transition X corresponds to task X.
- Places *Dormant* and $InProgress$ correspond to the states of the same name in section 2.1. The place *Discarded/Completed* corresponds to the termination of task X due to the satisfaction of its discarded or completed condition. While in the finite state transition system of a PRO*forma* task the places *Completed* and *Discarded* are connected with the place *Dormant* by arcs, we introduce the restriction that no incoming arc can be defined for any transition or place in the CPN corresponding to task X, except for the *Dormant* place.
- Conditions *Start(X), Discarded(X), Cycle(X),* and *Completed(X)* are interpreted as in section 2.1.
- Transitions *Start, Producetrigger, Discard* and *Complete* are introduced to allow change from one place to another when the corresponding conditions are satisfied.
- Only if the task X is in state *Dormant* transition *ProduceTrigger* can be fired changing the state of the task X to *InProgress*, independently of the scheduling constraints.

2. Algorithm to map PRO*forma* plans to Coloured Petri Nets

An arbitrary PRO*forma* task P of type plan can be mapped into a CPN where:

- The places *Dormant* and $InProgress$ are introduced, which are interpreted as explained in section 2.1. Two new places definitions called *Suspended* and *Terminated*

are introduced to represent the states where all the transitions are discarded due to the satisfaction of the discarded condition or completed condition, respectively. The place *Discarded/Completed* is introduced to represent the termination of the plan due to the satisfaction of the discarded or completed condition.

- The transitions *Start, Discard* and *Complete* are introduced and they are interpreted as explained in detail in [13]. These transitions correspond to the actions required to change the state of the plan P from *Dormant* to *InProgress*, from *Suspended* to *Discarded/Completed*, and from *Terminated* to *Discarded/Completed*. For each component of P two transitions *Terminate* and *Suspend* are added to change the state to *Terminated* and *Suspended*, respectively.
- Each component of plan P maps into a transition.
- The place *Dormant* is connected to the transition *Start*, which is conditioned by the predicate $Start(P)$ corresponding to the satisfaction of the start condition of plan P. Transition *Start* is connected to the place $InProgress$, which is connected to the transition $nStart$ where $n > 0$ corresponds to the number of initial activities of P.
- The place *Dormant* is connected to a transition *Suspend*, such that if the plan is discarded and can not start, then a token flows from place *Dormant* to *Suspended*.
- Transition $nStart$ is connected with each transition T_{X_j} through a place q_j. Where each T_{X_j} corresponds to an initial task X_j in P.
- For each task Y to which a task X is a subsequent task seven elements are introduced in the CPN: a place $q_{Y,X}$ that is interpreted as "Y has been completed", an arc connecting $q_{Y,X}$ with transition T_X, an arc connecting transition T_Y with $q_{Y,X}$, a transition *Terminate*, a transition *Suspend*, an arc from $q_{Y,X}$ to *Terminate* and an arc from $q_{Y,X}$ to *Discard*. The execution of task X is conditioned to the satisfaction of the predicate *cond(P,T_X)=notCompleted(P) and notDiscarded(P) and Start(X)*. Therefore task X can only be executed if task Y has been executed before, the plan P has not been completed or discarded, and its start condition is satisfied.
- Each transition T_X in the CPN, corresponding to a task X without outcoming transitions in P, is connected to the place $Terminated$.
- Each transition $Terminate$ ($Suspend$) has associated the condition $Completed(P)$ ($Discarded(P)$) corresponding to the satisfaction of the completion (discarding) condition of plan P. The transition $Terminate$ ($Suspend$) is connected to the place $Terminated$ ($Suspended$).
- The place $Terminated$($Suspended$) is used to collect the s (m) tokens, corresponding to the s (m) transitions which are pending for execution, when the completed (discarding) condition of the plan P is satisfied. The place $Terminated$ ($Suspended$) is connected with the transition $Complete$($Discard$) by an arc with cardinality s (m). Transition $Complete$ ($Discard$) has associated the condition $Completed(P)$ ($Discarded(P)$) that is satisfied when the plan P is completed (discarded). Only one token can flow from transition $Complete$($Discard$) to the place $Discarded/Completed$.
- The transition $Complete$ is connected with the places $InProgres$ and $Discarded/Completed$. If the condition *Cycle(P)* corresponding to the satisfaction of the cycling condition of plan P is satisfied, then the execution of $Complete$ fires one token to place $InProgress$. If the condition *notCycle(P)* corresponding to the non satisfaction of the cycling condition of P is satisfied, then the execution of $Complete$ fires one token to the place $Discarded/Completed$.
- The transition *Discard* is only connected with the place $Discarded/Completed$.
- For each transition T_X in P corresponding to a PROforma task X of type plan this algorithm can be applied.

– For each transition T_X in P corresponding to a PRO*forma* task X of type decision, enquiry or action algorithm 1 can be applied.

Example 1. In the right part of figure 1 we present the CPN corresponding to the plan *MedicalGuideline* presented in the left part of the same figure, where plan $MedicalGuideline$ is denoted as P.

3 Analysis of the expressiveness of a process-based language by workflow patterns

Now that we have a procedure for mapping any PRO*forma* plan to an equivalent CPN, we can apply formal analysis techniques developed in the PN domain. An interesting high-level property to investigate is the expressiveness of the mapped language. In [11] a set of 43 standard primitive workflow patterns is introduced (in CPN form) as a baseline for analyses. These have been used informally to compare CIG languages [7], but with our formal mapping we can now make this comparison more rigorous.

We can formally prove that a workflow language $\mathcal{L}$ satisfies a CPN pattern if we can define a process specification in $\mathcal{L}$ which can be mapped to a CPN with identical semantics.

First we exemplify this proof technique by showing that the PRO*forma* language satisfies pattern 10 of [11]. Pattern 10 specifies the ability to represent cycles in a workflow with more than one entry or exit point. Figure 3 shows the pattern with two entry points: $p3$ and $p4$. We will call this CPN *Pattern10*.

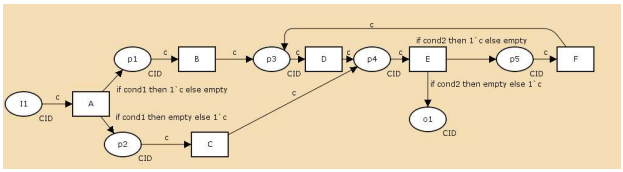

Fig. 3. Pattern 10

We prove that a PRO*forma* plan can be defined that specifies a cycle of this type with two entry points. The analysis can be generalized to an arbitrary cycle with $n \geq 0$ entry points.

In the PRO*forma* plan presented in figure 4 we assume that:

– The enquiry tasks *enquiry_1* and *enquiry_2* correspond to checking if condition *cond1* and *cond2* are respectively satisfied. These enquiries and the tasks $Start$, $Terminate$, $2Start$ and $Complete$ are internal tasks without external observable behavior.
– The precondition of task B is *cond1="yes"*, so task B can only be executed if *cond1* is satisfied. The precondition of task C is *cond1="no"*, so task C can only be executed if *cond1* is not satisfied.

– A cyclic plan, *PlanLoop*, is defined with attributes: *cycle_until= cond2="no", wait_condition= Completed(B) or Completed(C)* and *termination_condition= Completed(F) or (Completed(enquiry_2) and cond2="no")*. So plan *PlanLoop* can start its execution when tasks *B* or *C* are completed, *PlanLoop* cycles while *cond*2 is true and it is completed when *F* is completed, or *enquiry_2* is completed and *cond*2 is false. In *PlanLoop* task *D* is defined with *wait_condition= (Completed(B) and nro_cycle=1) or (nro_cycle>1)*, task *E* is defined with *wait_condition= (Completed(D) or (Completed(C) or nro_cycle=1))* and task *F* is defined with *precondition= cond2="yes"*.

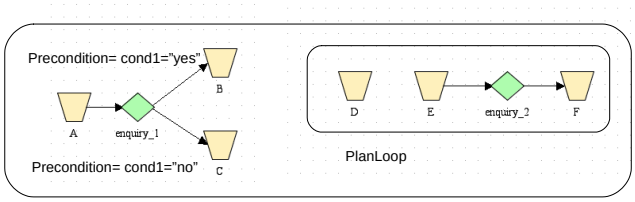

Fig. 4. PRO*forma* plan for Pattern 10

Applying the algorithm of section 2.3 to the plan from figure 4 we obtain the CPN of figure 5, that we call *PROf10*. Because in the plan from figure 4 no abort condition is specified, in *PROf10* there are no transitions *Suspend* and *Discard*, an there is no place *Suspended*.

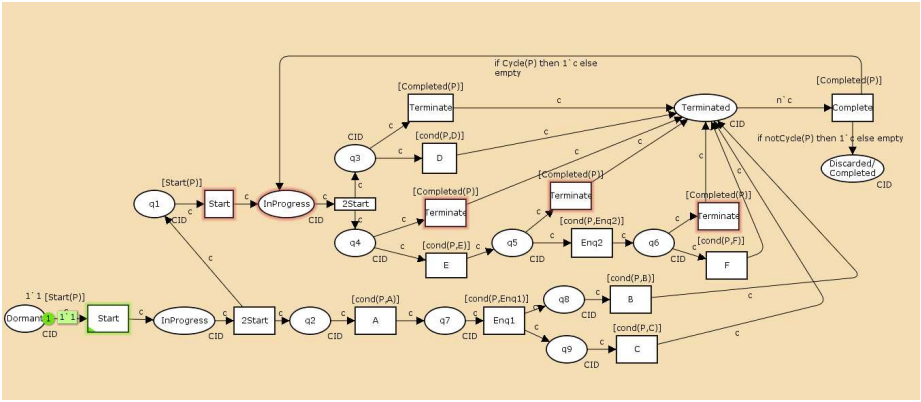

Fig. 5. CPN corresponding to the PRO*forma* plan presented in figure 4

PRO*forma* satisfies the pattern 10 because we can prove that:

1. *Pattern10 and PROf10 check conditions cond1 and cond2 at the same workflow point*: in both CPNs *cond*1 is inspected after the execution of transition *A*, and *cond*2 is inspected after the execution of transition *B*. Differences in the time of evaluation of conditions *cond1* and *cond2* in *Pattern10* and *PROf10* could lead to different workflows, because the execution of tasks *A, B, C, D, E* and *F* can change the truth value of *cond1* and *cond2*.

2. *Pattern10 and PROf10 generate the same observable behaviour:* given by the following regular expression: *AB(DEF)*DE + AC(EFD)* E*. The expression *AB(DEF)*DE* corresponds to the case in which condition *cond1* is satisfied. Then task *A* is executed before task *B*. While condition *cond2* is satisfied tasks *D*, *E* and *F* are sequentially executed. When *cond2* is not satisfied the workflow finishes executing tasks *D* followed by task *E*. The expression *AC(EFD)* E* corresponds to the case in which *cond1* is false.
3. *Pattern10 and PROf10 distinguish the same set of successfully terminated and deadlock processes:* in both CPNs the only deadlock processes are those in which the exit condition *cond2* of the cycle is never satisfied.

To prove that a workflow language $\mathcal{L}$ does *not* satisfy a pattern *PatternX* is harder because it requires proof that there is no composition of tasks in $\mathcal{L}$ that provides the behaviour defined by *PatternX*.

We will exemplify this proof technique by showing that PRO*forma* does not satisfy pattern 8 of [11]. This pattern specifies convergence of two or more branches of workflow into a single subsequent branch. Each enablement of an incoming branch results in the thread of control being passed to the subsequent branch. The left part of figure 6 illustrates the pattern with two incoming branches. We call this CPN *Pattern8*.

We prove here that PRO*forma* does not satisfy pattern 8 for the case of two incoming branches, but the proof can be generalized to $n \geq 0$ incoming branches. PRO*forma* does not provide a way to define task antecedents as disjunctions of predicates, therefore it is not possible to define for task *C* the antecedent tasks *Completed(A) or Completed(B)*. Therefore considering the PRO*forma* plan in the right part of figure 6, we have to consider three cases:

1- The plan is non-cyclic: this case is represented by plan *P1* from figure 6. PRO*forma* has a way to define task temporal constraints by the use of the attribute *task_antecedent*. But the kind of expressions that can be used for the definition of antecedent tasks are restricted to conjunctions of *Completed(X)* predicates, where *X* is an arbitrary task in the plan. In the case of task preconditions disjunctive predicates are allowed, therefore it could be possible to define as a *precondition* of task *C* the predicate *Completed(A) or Completed(B)*. But in this case task *C* would be discarded before it could be executed because no antecedent tasks can be specified. A *wait_condition* (*state trigger*) could not be used either. Although state triggers allow the definition of disjunctive predicates for conditioning the execution of task *C*, they can only activate task *C* once during the execution of the plan, and we must assume that the plan is non-cyclic. So even if we add the attribute *wait_condition= Completed(A) or Completed(B)* to task *C* , this task will never be executed more than once.

2- The plan is non-cyclic and task *C* is copied (cloned): as we show with plan *P2* from figure 6 we introduce two copies of task *C*. We specify antecedent tasks of the first copies the predicate *Completed(A)*, and as antecedent tasks of the second copy the predicate *Completed(B)*. In this case we can generate all the traces specified by Pattern 8. Pattern 8 requires that each execution of task *C* follows a unique thread of control. But there is no way to merge the two different threads of control opened by the execution of the copies *C*, as it was explained in the previous case.

3- The plan is cyclic with two cycles: this case is represented by the plan *P1* from figure 6. From the previous analysis it is clear that the only option to be considered is

to define for task C the attribute *wait_condition= Completed(A) or Completed(B)*. But then the trace language generated includes the traces $ACAC$, $BCBC$, because there is no way to restrict the plan such that if in the first execution A (respectively B) was performed then in the second execution only B (respectively A) can be executed.

We conclude that there is no PRO*forma* plan that can simulate pattern 8.

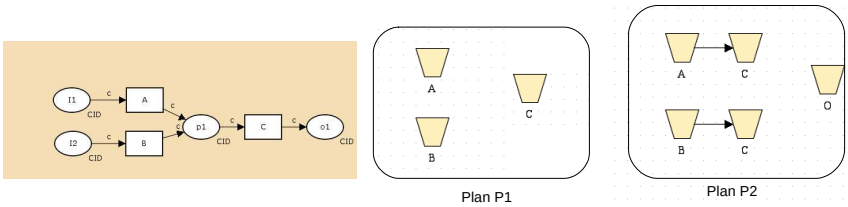

Fig. 6.

4 Conclusions

From the algorithm of section 2.3 and the analysis of PRO*forma* expressiveness in section 3 we deduce that PRO*forma* cannot express arbitrary PNs (i.e. $PROforma \subset PN$).

In [7] it is argued informally that PRO*forma* satisfies 23 of the 43 baseline patterns of [11]. The formal approach demonstrated here allows a more rigourous analysis of expressiveness, and we expect that this would yield a somewhat different subset of patterns (indeed [7] has PRO*forma* unable to express pattern 10, which we have shown here it is able to express, although we agree with [7] that pattern 8 cannot be expressed in PRO*forma*).

However the more important and more general point is that such an analysis can pinpoint exactly which features of a language reduce its expressiveness with respect to a particular workflow pattern, and can provide well grounded arguments for adding features to a language in a principled, rather than ad-hoc, way. For example:

1- In PRO*forma* it is only possible to define task temporal constraints as conjunctions of Completed(X) predicates, where X is a task in the plan. A future PRO*forma* extension should allow the use of disjunctive predicates in the specification of task temporal constraints.

2- PRO*forma* does not provide persistent internal triggers. Adding internal persistent triggers (retained by the workflow until they can be acted on by the receiving activity) would allow a task to be triggered by a signal from another part of the plan. If in the non-cyclic PRO*forma* plan $P1$ shown in figure 6 we associate an internal persistent trigger *PerformC* with task C, and we define as postconditions of tasks A and B the predicate *triggered(PerformC)*, then we obtain a PRO*forma* plan that can simulate the pattern *Pattern8*.

As future work we plan to study the patterns not satisfied by PRO*forma* in order to propose a minimal extension that provides PRO*forma* with the expressive power required to satisfy all the patterns from [11]. We are also interested on exploring how to enrich PRO*forma* with notions of time based on time PNs

and notions of multithreaded execution based on CPNs, and to investigate formal analysis of structural, temporal and behavioral properties of PRO*forma* guidelines based on the automatic model checking tools available for PNs.

References

1. A. Agostini, G. De Michelis and K. Petruni, Keeping Workflow Models as Simple as Possible, *Proceeding of the workshop on CSCW, Petri Nets and related formalism*: 11-29, 1994.
2. B. Chen, G.S. Avrunin, E.A. Henneman, L. A. Clarke, L.J. Osterweil and P.L. Henneman, Analyzing medical processes, *ICSE '08: Proceedings of the 30th international conference on Software engineering*: 623-632, 2008.
3. C.A. Ellis and G.J. Nutt, Modelling and Enactment of Workflow Systems, *Application and theory of Petri Nets LNCS*, 691: 1-16, 1993.
4. A.X. Garg, N.K.J. Adhikari and H. McDonald, Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: a systematic review, *JAMIA* 293: 1223-1238, 2005.
5. J.M. Grimshaw and I.T. Russel, Effect of clinical guidelines on medical practice: A systematic review of rigorous evaluations, *Lancet* 342: 13171322, 1993.
6. B. Kiepuszewski, *PhD Thesis: Expressiveness and Suitability of Languages for Control Flow Modelling in Workflows*, Faculty of Information Technology, Queensland University of Technology, 2003.
7. N. Mulyar, W.M.P. van der Aalst, M. Peleg, A pattern-based analysis of clinical computer-interpretable guideline modeling languages, *JAMIA* 14(6):781-7, 2007.
8. C.A. Petri, Kommunikation mit Automaten, Bonn: Institut fr Instrumentelle Mathematik, Schriften des IIM Nr. 2, 1962.
9. F. Puhlmann, Why Do We Actually Need the Pi-Calculus for Business Process Management?, *Business Information Systems* 2006: 77-89, 2006.
10. S. Quaglini, M. Stefanelli, A. Cavallini, G. Micieli, C. Fassino and C. Mossa, Guideline-based careflow systems, *Artif Intell Med*: 20(1), 5-22, 2000.
11. N. Russell, A.H.M. ter Hofstede, W.M.P. van der Aalst, and N. Mulyar, Workflow Control-Flow Patterns : A Revised View, *BPM Center Report BPM-06-22*, 2006.
12. I. Salomie, T. Cioara, I. Anghel, M. Dinsoreanu and T.I. Salomie, Workflow models enhanced with process algebra verification for industrial business processes, *Proceedings of the 11th WSEAS International Conference on Computers* 502-507, 2007.
13. D.R. Sutton and J. Fox, The Syntax and Semantics of the PRO*forma* guideline modelling language, *JAMIA* 10(5):433-43, 2003.
14. A. ten Teije, M. Marcos, M. Balser, J. van Croonenborg, C. Duelli, F. van Harmelen, P. Lucas, S. Miksch, W. Reif, K. Rosenbrand and A. Seyfang, Improving medical protocols by formal methods, *Artificial Intell. in Medicine* 36(3):193-209, 2006.
15. A.H.M. ter Hofstede and E.R. Nieuwland, Task structure semantics through Process Algebra, *Software Engineering Journal*, 8 (1): 14-20, 1993.
16. W.M.P. van der Aalst, K.M. van Hee and G. J. Houben, Modelling and analysing workflow using Petri Net based approach, *Proceedings of Second Workshop on Computer-suppported Cooperative work, Petri nets related formalisms*: 31-50, 1994.
17. W.M.P. van der Aalst, The Application of Petri Nets to Workflow Management, *The Journal of Circuits, Systems and Computers*, 8(1), 21-26, 1998.

Management of Knowledge-intensive Healthcare Processes on the Example of General Medical Documentation

Danny Ammon[1], Dirk Hoffmann[2], Tobias Jakob[2], Ekkehard Finkeissen[2], Vesko Detschew[1], Thomas Wetter[3]

[1] Technische Universität Ilmenau, Gustav-Kirchhoff-Str. 2, 98693 Ilmenau, Germany
[2] medrapid GmbH, Schweizertalstr. 35, 69118 Heidelberg, Germany
[3] Universitätsklinkum Heidelberg, Im Neuenheimer Feld 672, 69120 Heidelberg
{danny.ammon, vesko.detschew}@tu-ilmenau.de
{dirk.hoffmann, tobias.jakob, ekkehard.finkeissen}@medrapid.info
thomas.wetter@med.uni-heidelberg.de

Abstract. Healthcare Processes are characterized by knowledge-intensive tasks. In contrast to this, most of the efforts for business process management in healthcare do not refer to this quality, and software engineering in healthcare relies on an unspecific process-oriented approach. In this contribution, we present a method of capturing and analysis of a knowledge-intensive process, from which we derive requirements to a knowledge- as well as process-oriented information system for the example of general medical documentation. We present the resulting implementation of a knowledge-based electronic patient record and discuss the potentials and open issues for our proposal.

Topics: process modeling in healthcare, process-oriented system architectures in healthcare, integrating healthcare processes with electronic medical records

1 Introduction

1.1 Motivation

Successful approaches of process management in the branches of healthcare as well as in any other sector will have to reflect the special characteristics of the domain they deal with. For healthcare this is the medical domain, applied in everyday patient treatment. The domain of medicine is very complex, subdivided into many specialized fields, and is not only changing rapidly, but also growing exponentially. Furthermore, the demand of up-to-date medical knowledge in patient supply is highly critical in terms of time. We can therefore classify healthcare as a knowledge-intensive sector.

Knowledge utilized in patient treatment can at least be divided into three types:

1. *General medical knowledge* is a result of research and independent of individuals.
2. *Institutional knowledge* refers to the properties and functions of the special healthcare institution where a clinician is acting.

3. *Patient-specific knowledge* is generated during the consultation and the diagnostic measures of the clinician.

It is the task of every professional working in the sector of healthcare, to map general medical knowledge onto the information gathered about a specific patient, and to combine this with the institutional knowledge to determine the optimal diagnostic or therapeutic procedures for the ongoing process of treatment.

The leading questions for this contribution are now: How can the knowledge intensity of healthcare be taken into account for a suitable business process management in healthcare? How is it possible to add *process knowledge* to the institutional knowledge of the clinician, where this important role and use of knowledge in the processes themselves is incorporated? What are the advantages and potentials of a knowledge-oriented business process management in healthcare? And, last but not least: Can these potentials be utilized for deriving requirements and specifications of an adequate process- and knowledge-oriented IT support? How could such an information system look like? We will consider these questions here by the exemplary process of general medical documentation and the associated information system, the electronic patient record.

1.2 Overview of the Contents

In this contribution, we define a knowledge-oriented healthcare process management approach using the example of general medical documentation. We begin by providing a background and reviewing related work, including the suggestion of an appropriate language for the modeling of knowledge-intensive business processes, the Knowledge Modeling and Description Language (KDML). This language is then used for establishing exemplary and generic models of patient treatment including documentation (process model), and of clinical documentation itself (activity model).

In the next step, out of the models we present, requirements for a knowledge-oriented information system for the process of clinical documentation are derived. As a possible result of our approach, we present a knowledge-based electronic patient record and its characteristics in the subsequent paragraph.

Finally, we discuss the potentials as well as open questions for the here-proposed type of process management and information system.

2 Background and Related Work

2.1 Knowledge in Healthcare

The analysis of medical knowledge is a research topic both of theoretical medicine and of healthcare knowledge management. Additional and more profound differentiations of types of medical knowledge, have been accomplished there [4][11][1]. The resultant *knowledge dilemmas* in medicine have been expressed similarly: At first the impossibility for a modern clinician even in medical subareas to access all available general scientific cognitions right at the moment of a clinical decision in a special case [1]. Moreover, the difficulties which arise from the

appliance of abstract knowledge to individuals [19]. An associated characteristic is the retroactive effect of the invention of new therapeutic options on the differentiation of disease patterns which adds to medicine's exponential growth [6].

An additional classification of knowledge in general has been established by Polanyi, who distinguished between formal, verbalized, communicatable, *explicit knowledge* and human-bound, experiential, believed, sensed, *tacit knowledge* [15]. Both types of knowledge play an important role in healthcare, explicit knowledge as general textbook knowledge and tacit knowledge as clinical experience both affecting every clinical decision.

Management of knowledge in companies, as a different subject, will have to rely on the mentioned findings on general and medical knowledge. An accordant approach, the well-known model of the dynamics of knowledge creation has been developed by Nonaka and Takeuchi [13], which in turn influenced the modeling language we suggest for healthcare process management later in this and in the following section. They divided the conversion of knowledge in business processes in four types: *internalization*, which creates tacit knowledge, *externalization*, which creates explicit information objects, *socialization* which refers to the communication of tacit knowledge, and *combination* which means deriving new information from the use of two or more information objects.

2.2 Formalization of Medical Knowledge

Explicit medical knowledge as a part of medicine has been subject to various kinds of formalization, with or without IT support. Medical *terminologies*, such as the ICD [22] or SNOMED [17] have been developed for coding, statistical analysis, reimbursement or decision support. A representation of medical semantics and clinical information [3] has been achieved through the creation of medical *ontologies*, e.g. UMLS [21] or OpenGALEN [14]. Finally, the application of medical knowledge bases for encyclopediae, didactics, as terminology server or as decision support system, has been established as a result of medical knowledge engineering. An example for a modern knowledge-based system is medrapid, a professional healthcare internet portal for the distribution and exchange of general medical information in health care [5]. The medrapid knowledge base serves as fundament for the exemplary application of the findings in this paper to a process-oriented, knowledge-based information system for general medical documentation, in the fifth section.

2.3 Models of Clinical Practice

Not only medical knowledge, but clinical practice as well has been subject to analyses and modeling. The generic model of medical treatment which is shown in the next section, relies on the proposal made in [12]. The author distinguishes two loops of acting of a clinician: The *diagnostic loop* which is repeated until the findings for a certain patient are sufficient to deduce a diagnosis, and the therapeutic loop which is repeated until a special therapy for the given diagnosis has proven effective. This model of the clinical action is very abstract and idealized (what happens if a given

diagnosis has to be revised because of therapeutic findings?) but shall serve as a basis for the simplified example proposed in this contribution. An overview and assessment of generic models of clinical practice and their and direct utilization for the design of electronic patient record systems can be found in [18].

2.4 Modeling Languages for Knowledge-intensive Business Processes

A first step in the management of business processes is their identification and documentation. For this task, various process modeling languages have been developed. In the last years, several approaches of extensions or additions to these languages have been introduced, which allow a knowledge-oriented analysis of processes [9]. Furthermore, there are several types of healthcare modeling languages used for documentation of clinical practice guidelines or clinical pathways [1].

Out of these options, for this contribution we chose the general knowledge-process modeling language KMDL (Knowledge Modeling and Description Language) for the following reasons: Traditional business process modeling languages do not or only in a static view regard knowledge objects in business activities [8]. Healthcare process modeling languages, on the other hand, directly represent medical knowledge but do not link it to objects or subjects and, in case of clinical practice guidelines, do not incorporate institutional knowledge [1].

Knowledge process modeling languages like KMDL focus instead on knowledge types, on conversion of knowledge in the context of business processes and therefore provide a link between the utilization of domain-specific knowledge as it is represented in medical guidelines, and the illustration of general business processes as well as healthcare-specific activities. KMDL is directly based on the knowledge management approach by Nonaka and Takeuchi [7][8]. The current KMDL version 2.1 provides two perspectives onto the sequence of actions, where the *process view* shows the business process as an execution of single tasks, whereas the *activity view* is used to consider the knowledge conversions during the fulfilling of a special tasks. The defined entities for modeling of a process view as well as an activity view can be found in fig. 1.

3 Modeling Knowledge-Intensive Healthcare Processes

In this section, we use the KMDL approach of capturing knowledge-intensive business processes for the design of a generic model which focuses on clinical treatment as a whole in the process view, whereas in the activity view we focus on the general medical documentation as a type of knowledge conversion during the treatment process.

We concentrate on clinical documentation as special knowledge-intensive process because of the fact that every clinical action is the result of an *interpretation* of data or information by the clinician. This generation of knowledge by interpretation itself is one of the main reasons of the legal obligation for documentation in healthcare in many countries. Therefore, the fundament of the activity of documentation is clinical

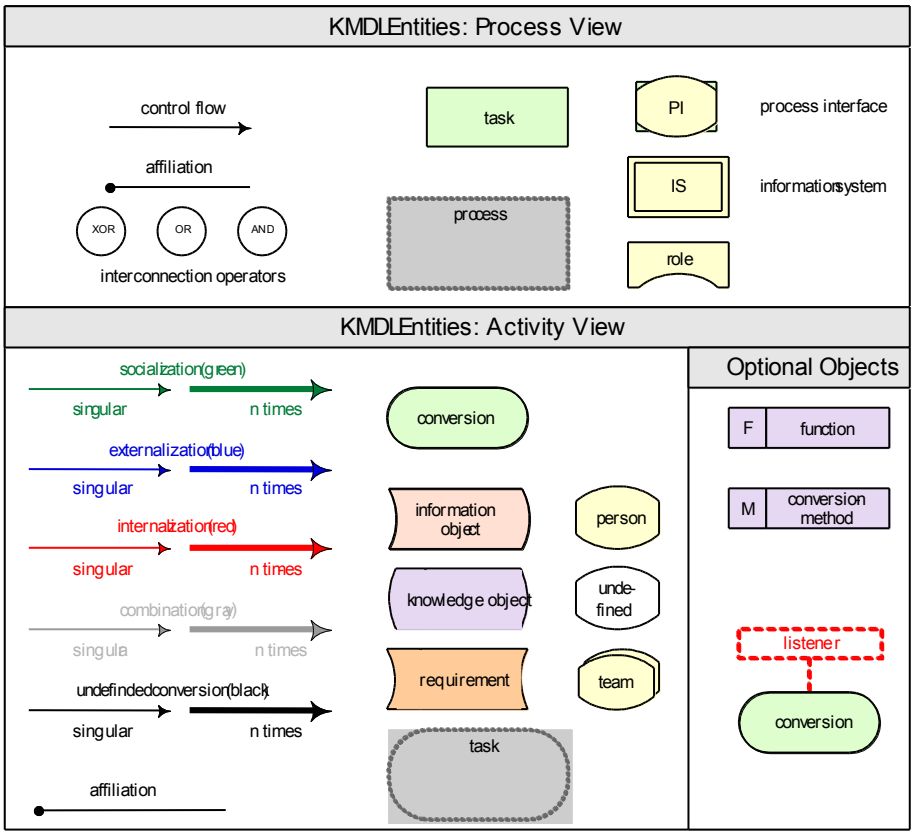

Fig. 1. KMDL Entities for Process View (above) and Activity View (below).

knowledge and its characteristics itself, distinguishing at as one of the basic activities directly and completely referring to knowledge in healthcare as we defined it.

3.1 Process View of Clinical Treatment

The first approach of characterizing clinical documentation is a generic model of clinical treatment itself, which provides information about the specific time when documentation is necessary as well as the concrete circumstances which are to be documented. A KMDL process view of clinical treatment, based on the statements in [12] is shown in fig. 2.

Roles have been omitted in the process view since for this idealized model there is only one actor—the clinician. The relevant information system for the further consideration is the electronic patient record, which has been assigned to every task of documentation.

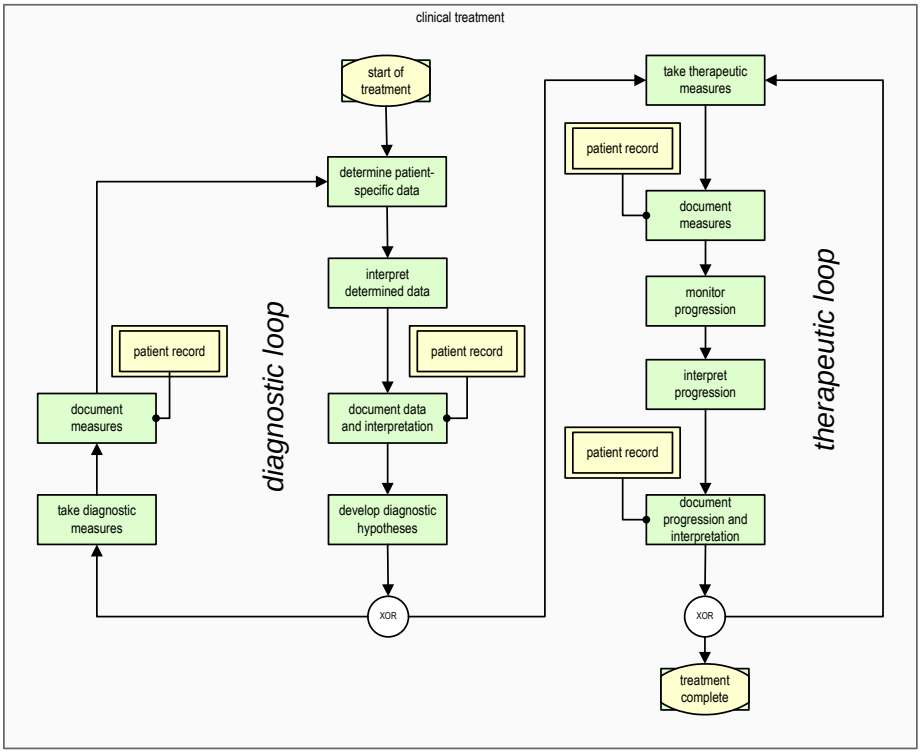

Fig. 2. KMDL Model of Generic Medical Treatment (Process View)

We can now derive the following assumptions from this abstract model:

- general medical documentation is related to diagnostic and therapeutic actions
- two types of documentation can be found, affecting on one hand concrete data or executed measures, on the other hand interpretations of the clinician (to which the diagnosis itself belongs as well)

The process model has thus already been helpful in analyzing the documentation process; in the following we will deduce further assumptions from the activity view.

3.2 Activity View of Clinical Documentation

In fig. 3 the model of clinical documentation is presented as KMDL activity view. We now introduce the role or person clinician, which utilizes as least the three different knowledge objects as we mentioned them, for the externalization of his tacit knowledge about medicine in general, the institution he or she is working in, and about the special patient he is treating at the moment. At the same time, this knowledge as a whole is a requirement for the correct completion of the documentation task: One who lacks medical knowledge, is not aware of the

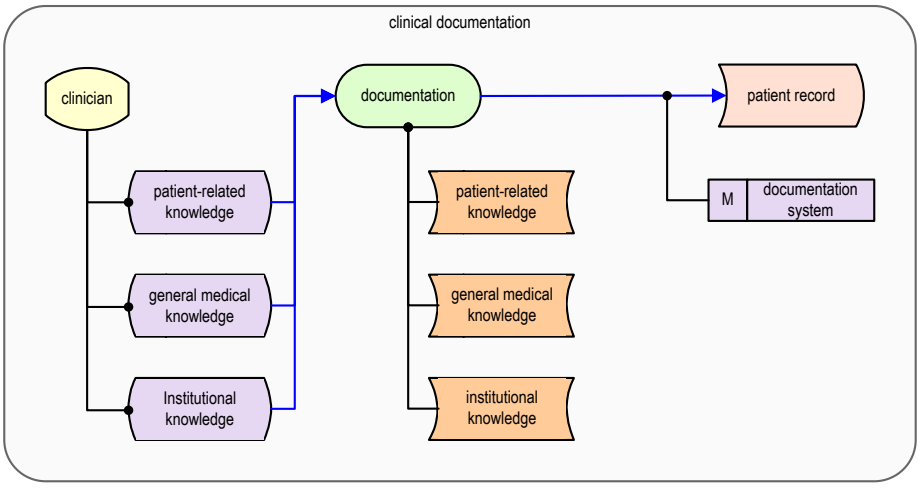

Fig. 3. KDML Model of General Medical Documentation (Activity View)

characteristics of his or her institution, or does not know the patient, cannot correctly document any finding, action or interpretation.

So which findings can we derive from the activity model?

- three types of knowledge are required for clinical documentation
- both data resp. measures and interpretations are externalized with this knowledge in background
- the knowledge included in a patient record is individual-bound to the treating clinician

The KMDL model of clinical documentation has therefore provided additional insights to the actual process, based here only on a generic approach which contains no institutional details or deviations.

4 Deriving Requirements and Potentials for Supporting IT

In the previous section, we have documented and analyzed a generic healthcare process by means of knowledge-oriented business process modeling. In this chapter, we will continue the short example by transforming the resulting findings into requirements to an information system designed to support the process. Since we chose clinical documentation, the information system needed is an electronic patient record where the knowledge the clinician externalizes to is case-specifically structured.

Another necessary structure in the resulting system has to be the special date of treatment (called session), based on the sequential or time-specific tasks the process model is built of. Furthermore, when we look at the process model as a whole, we

find this long-term process describes what clinicians call an episode of care—"a series of temporally contiguous health care services related to treatment of a given spell of illness or provided in response to a specific request by the patient or other relevant entity" [10]. An electronic patient record designed on the basis of the knowledge-oriented process models we presented, must therefore structure the inserted data patient-, episode- and session-based.

Directly adopting from the assumptions based on the model, the supporting system has to allow the classification of certain diagnostic as well as therapeutic measures and findings. Furthermore, personal interpretations of a clinician will have to be marked as such, including a diagnosis.

Specifically, the potential of an electronic patient record lies in the support of general, patient-independent knowledge that is relevant to clinical documentation. Such general knowledge need not be individual-bound and can therefore be stored as explicit knowledge in the information system itself. This change in knowledge-utilizing during the process of documentation can avoid errors prevailing whenever a single person has to rely on his or her own mind for the externalization of actually already explicit, general knowledge.

In the following section, we propose an information system for general medical documentation which directly derives from the here-mentioned knowledge-oriented requirements of the documentation process.

5 Example: Knowledge-based Clinical Documentation

5.1 Specifications of a Knowledge-oriented Documentation System

In this section, we present the specifications of an electronic patient record which convert the above-developed assumptions and requirements concerning the process of clinical documentation in general.

Starting with the proposed potential of a system which provides knowledge support, the electronic patient record we suggest is knowledge-based. There are two potential types of knowledge which can be stored in the system, the more complex one being general medical knowledge, and the one referring to this knowledge being institution-specific knowledge. Supporting functions result in certain input guidelines which, for example, suggest diagnostic measures for a certain kind of suspected diagnosis and, out of these, highlight the ones which can be executed in the institution the clinician works in.

Furthermore, we distinguish between four structuring concepts as components of the software:

1. The *patient*, which forms the upper classification of data since all documentation data is patient-specific.
2. The *episode* which is described by the process of treatment shown in the process model; when this process starts again for a certain patient, a new episode begins.
3. The *session* as time-specific structuring criterion which keeps the option of a chronological view at the record.

4. The *clinical condition* resp. the *executed measures*. While the clinical condition represents the clinician's interpretation on the information provided by the patient, the executed measures represent explicit knowledge about the actions implemented during treatment, including and differentiating diagnostic as well as therapeutic measures.

In the following, we present details of a reference implementation of these specifications.

5.2 Electronic Patient Record Based on medrapid

The above-gathered specifications of a knowledge-based electronic patient record have been implemented as a documentation system based on the medrapid knowledge base [2]. Since medrapid is an institution-independent web portal, there is no *institutional knowledge* stored up to now. medrapid allows searching, navigating and communicating of around 9,000 disease patterns, 47,000 diagnostic measures and 16,000 therapeutic alternatives. An example of a clinical condition referring to the hepatitis disease is shown in the screenshot in fig. 4.

The electronic patient record relating to this knowledge base has been designed and implemented utilizing the CommonKADS methodology for knowledge-based applications [16], which has also shown to be suitable for capturing knowledge-intensive business processes [20]. The resulting web-based application consists of three components: a patient table, a session and episode diagram, and the file view showing clinical conditions and executed measures for the chosen patient at the chosen time. Further details on the architecture of the software are provided in [2], here the emphasis lies on the implementation of the model-based specifications: As a visual example, the knowledge-based documentation of a clinical condition is shown in the tree-type view in fig. 5. The *general medical knowledge* of hepatitis is adopted

Fig. 4. Example of General Medical Knowledge Stored in medrapid

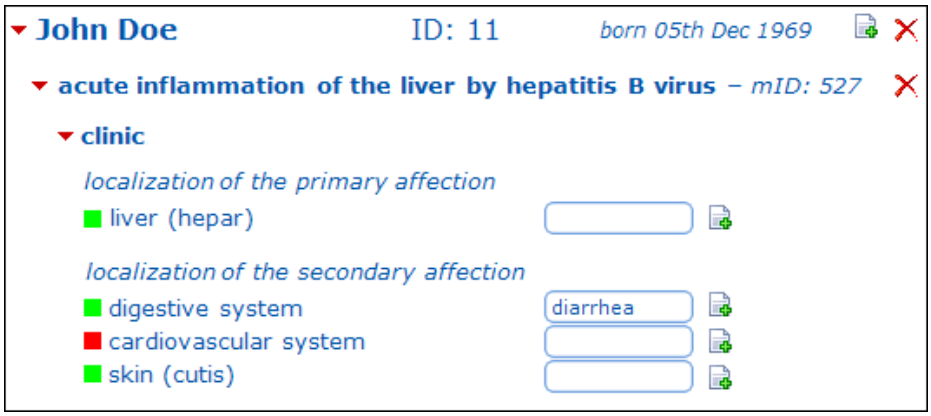

Fig. 5. Possible User Interface of a Knowledge-based Electronic Patient Record

from the knowledge base, suggesting a superset of possible clinical conditions, from which the clinician has to specify the patient's condition (*patient-related knowledge*) by activating the suggested items (red / green square before the item). Furthermore, additions to the active items are possible in free text fields, building a set of patient-specific knowledge independent from but linked to the general medical knowledge. In the exemplary screenshot, the clinician can specify the localization of hepatitis for the patient, e.g. to describe the current status of the liver as primarily affected organ, or to insert special secondary affections

As a third way of documentation, additional files or documents belonging to a patient's clinical condition resp. diagnostic or therapeutic measures, can also be uploaded (document icon behind the text field).

The resulting application in all is thus designed as a proposal for an episode-oriented, knowledge-based electronic patient record implemented according to the specifications derived above.

In the following last section, we summarize the results of our research, raise open questions and develop possible approaches for future tasks in this field.

7 Conclusions and Future Work

In this contribution, we have used a generic medical process as an simple example to show the possibility of a knowledge-oriented healthcare process management approach. We have proposed a KMDL process and activity view of the clinical documentation and concluded several statements from the analysis of these models. We were able to transform these statements into requirements to a knowledge-oriented optimization of clinical documentation, and have taken these requirements as a basis for the specification of documentation-supporting information system. Finally,

we have presented the medrapid knowledge-based electronic patient record as an exemplary implementation of the established specifications.

The process example is of course a short and abstract one, which does not refer to a concrete, institution-specific business process referring to specific personnel, activities, or surroundings. As a result, institutional knowledge has been omitted from the specifications and the implementation, keeping the approach a generic one. Furthermore, general medical documentation, too, is only a part of real documentation processes, which also must refer to or initiate additional administrative processes like patient transport or drug order, and in some cases are supplemented by very specific, technically-detailed documentation types like an operation summary, which bases on a very detailed, new quality of medical knowledge.

A good start for future work is therefore the application of the proposed method to more practical process examples, where two approaches alone—business process modeling and process-oriented software-engineering—have already proven effective. Besides that, the spectrum of process analysis must be broadened from clinical treatment resp. general medical documentation alone to a variety of other clinical processes, including administrative and care activities.

Additionally, the knowledge-based electronic patient record as it was implemented, has yet to be practically used to examine in detail the effect of a knowledge support to the documenting clinician. This could be possible by an evaluation period where "traditional" electronic methods of general medical documentation are compared to the knowledge-based method in terms of effectiveness and efficiency.

The completion of the suggested future tasks would provide a good support for the preliminary unified approach presented here, which leads from a capture of knowledge-intensive processes in healthcare over the generation of requirements for an optimized variant of these processes to the transformation into details of process- and knowledge-oriented information systems. The resulting software has to be best-suited for the support of clinicians executing time-critical, knowledge-intensive duties in their everyday clinical work.

References

1. Abidi, S.S.R.: Healthcare Knowledge Management: The Art of the Possible. In: Riaño, D. (Ed.): K4CARE 2007. LNAI 4924, pp. 1–20. Springer, Berlin, Heidelberg (2008)
2. Ammon, D. et al.: Developing an Architecture of a Knowledge-Based Electronic Patient Record. In: Proceedings of the Thirtieth International Conference on Software Engineering, pp. 653–660. ACM, New York (2008)
3. Beale, T., Heard, S.: An Ontology-based Model of Clinical Information. In: Stud. Health Technol. Inform. 129, 785–90 (2007)
4. Doroszewski, J.: Unity and Diversity of the Medical Action: A Review of its Components and Their Interconnections. Metamed. 2, 155–168 (1981)
5. Finkeissen E., et al.: medrapid: Medical Community & Business Intelligence System. In: Health Data in the Information Society: Proceedings of MIE2002. IOS Press, Amsterdam (2002)
6. Gode, A.: Just Words. J. Amer. Med. Ass. 205, 352 (1968)
7. Gronau, N., Müller, C., Korf, R.: KMDL: Capturing, Analysing and Improving Knowledge-Intensive Business Processes. J. Univ. Comp. Sci. 4, 452–472 (2005)

8. Gronau, N., Weber, E.: Management of Knowledge Intensive Business Processes. In: Desel, J., Pernici, B., Weske, M. (Eds.): BPM 2004. LNCS 3080, pp. 163–178. Springer, Berlin, Heidelberg, 2004
9. Hädrich, T.: Contrary Positions About Modeling Knowledge Work. In: Althoff, K.-D. et al. (Eds.): WM 2005. LNAI 3782, pp. 248–258. Springer, Berlin, Heidelberg (2005)
10. Hornbrook M., Hurtado A., Johnson R.: Health Care Episodes: Definition, Measurement and Use. Med. Care Rev. 42, 163–217 (1985)
11. Hucklenbroich, P.: Klinisch-methodologische Aspekte medizinischer Expertensysteme. In: Hucklenbroich, P., Toellner, R.: Künstliche Intelligenz in der Medizin, pp. 9–36. Gustav Fischer, Stuttgart, Jena, New York (1993)
12. Hucklenbroich, P.: Wissenschaftstheorie als Theorie der Medizin: Themen und Probleme. In: Deppert, W. et al. (eds.): Wissenschaftstheorien in der Medizin: Ein Symposium, pp. 65–96. de Gruyter, Berlin (1992)
13. Nonaka, I., Takeuchi, H.: The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press, New York (1995)
14. OpenGALEN, http://www.opengalen.org/
15. Polanyi, M.: Personal Knowledge: Towards a Post-Critical Philosophy. The University of Chicago Press, Chicago (1958)
16. Schreiber, G., et al.: Knowledge Engineering and Management: The CommonKADS Methodology. MIT Press, Cambridge, MA (2000)
17. Systematized Nomenclature of Medicine, http://www.snomed.org/
18. Tange, H.J., Dietz, J.L.G., Hasman, A., de Vries Robbé, P.F.: A Generic Model of Clinical Practice: A Common View of Individual and Collaborative Care. Methods Inf. Med. 42, 203–211 (2003)
19. Thomasma, D.C.: Applying General Medical Knowledge to Individuals: A Philosophical Analysis. Theor. Med. 9, 187–200 (1988)
20. Trier, M., Müller, M.: Towards a Systematic Approach for Capturing Knowledge-Intensive Business Processes. In: Karagiannis, D., Reimer, U. (eds.): PAKM 2004. LNAI 3336, pp. 239–250. Springer, Berlin, Heidelberg (2004)
21. Unified Medical Language System, http://www.nlm.nih.gov/research/umls/
22. WHO International Classification of Diseases, http://www.who.int/classifications/icd/en/

Promoting Process-Based Collaborative Awareness to Integrate Care Teams

Federico Cabitza, Marco P. Locatelli, and Carla Simone

Università degli studi di Milano-Bicocca, Milano I-20127, Italy,
{cabitza,locatelli,simone}@disco.unimib.it

Summary. A number of literature contributions illustrate how collaborative awareness can improve process coordination in distributed work settings. The paper discusses this theme in the light of distributed care supported by Integrated Care Pathways. These can be used as relevant source of information to propagate this kind of awareness information. The paper shows on a reference scenario how the CASMAS model can support the design of collaborative applications by focusing on awareness promotion: this can be fostered and modulated to reduce information overflow by using specific features of the model.

Key words: Clinical Pathways, Process-Based Awareness, Modulated Awareness, Flexibility

1 Motivation

In the last decades, the increasing need for healthcare quality improvement and resource use optimization has led to the re-organization of care delivery processes within healthcare organizations according to the tenets of the "continuity of care" and "patient-centered care protocols". This is especially true in those cases where complex health problems require close collaboration between practitioners of different competencies and disciplines. In this view the patient is at the cross of different caring trajectories [17]: each trajectory owns its caring network and work flow and each network is responsible of specific interventions and the related information. The consequent multitude of specialized competencies and behaviors is the main cause of the care fragmentation that healthcare organizations are still experiencing as well of their difficulty in integrating the number of medical specialties and specialized departments, which is increasing along with the progressive refinement of medical treatments and techniques.

Even in facilities where all the relevant pieces of information about a single patient are shared and stored in a common and accessible documental base (be it either paper-based or computer-based), what it is still missing is a *pragmatic reconciliation* of how actors within specific groups see their local partitions and combine these with other partitions of the overall care process. With pragmatic reconciliation we mean a reconciliation about 'what to do next', i.e., about the proper use of information that members of different teams make to inform and affect their actions and decision making.

As researchers active in the field of CSCW, we are aware of observations and field studies claiming that clinicians can reach effective coordination and seamless cooperation without imposing or even proposing them any virtuous or standard flow of work. Specifically, medical work, with all its uncertain variables and unpredictabilities, requires the use of flexible means that are able to follow their users' needs in the most unobtrusive way. The CSCW research has collected evidences that one of such means is *collaborative awareness* [15]. In fact, it is usually the case that collaboration improves when people not only share information bul also can actively produce and maintain an idea of what is going on around them [13] [4].

Usually collaborative awareness is taken for granted as an aspect of those work settings where collaboration occurs face to face. Instead, in domains where people work in an asynchronous, document-mediated and distributed manner, this natural and "cheap" way to coordinate each others almost disappears. In these cases, information technology plays an essential role in leveraging the innate capability of workers to coordinate with their colleagues in ad-hoc, local and informal ways. In the healthcare domain, even if the purpose of any patient record is to record information about the patient, there are evidences that practitioners seldom look for information about the patient per se, but rather for information about the activities of other health-care workers regarding that patient (e.g., [14]). Traditionally, clinicians stay aware of each other's activities through informal and unanticipated interactions, like when they talk, while they are reviewing data in hand-over conferences or when they are ordering drugs and interventions at the patient's bedside. These interactions both raise awareness and provide incentives for members of the team to interact in significative ways [13].

These effective modes of interaction should be preserved and even fostered by any form of digitization starting from the introduction of the electronic patient record (EPR) [7]. In the same line, our focus is on a series of artifacts and process maps, usually called either *clinical* or *integrated care pathways* [3], that incorporate the procedural and articulative knowledge that clinicians externalize and consult on their own accord during patient care. This process-oriented knowledge about how activities should be articulated within and across responsibility borders is reified in a process model that constitutes a common reference for all actors involved in the related care trajectories. Our point is that these artifacts are a valuable support for the integration of information and knowledge within and across borders between different disciplines, competencies and responsibilities, thus reducing the risk of care fragmentation in concrete terms. Specifically, our proposal is to use integrated care pathways (ICP) as support for the promotion of that collaborative awareness that realizes the pragmatic reconciliation mentioned above.

In the next sections we illustrate the basic tenets of the model, called CASMAS (Community Aware Situated Multi Agent System), which we propose to design systems supporting the cooperation of actors involved in ICP of vari-

ous kinds, and illustrate its use in a reference scenario. The conclusive section delineates our future work.

2 Integrated Care Pathways and the CASMAS model

An ICP is a structured multidisciplinary care plan which outlines essential steps in the care of patients with a specific clinical problem [9]. ICPs are models of care intended to support the distributed and yet integrated management of an effective and efficient treatment of similar patients by practitioners of different competencies and responsibilities. By referring to a single ICP and to the local and national guidelines they incorporate, practitioners are supported in reducing unnecessary variations, in anticipating outcomes and coordinative handovers with external services and in developing care partnerships with their colleagues also across borders between specialities and facilities [11]. Since clinicians must be left free to meet the contingent needs of individual patients even in routine cases, they take ICPs in a flexible and descriptive manner and as a tool to reconciliate and acknowledge necessary variations across organizational borders. Yet, ICPs usually fall short to reach this particular aim whenever they are not supported by an information infrastructure that keeps different organizational units connected and aligned with the common process model.

The process models representing ICPs contain some basic kinds of information, irrespectively of the specific language or notation used to express them (e.g., ProForma [8] and GLIF [12]). First of all, ICPs represent the main caring activities and the causal relations between them: these relations indicate a partial order representing sequential, concurrent and alternative behaviors, possibly complemented by time related information (e.g., duration, frequency or number of iterations). Secondly, ICPs express the resources involved in those activities: they can be pieces of information, typically but not exclusively derived from the EPR, or instrumental resources. Finally, they specify the roles or organizational units that are responsible for single activities.

In this view, a patient is at the center of pathways that can involve several roles or different pathways in charge of a specific caring trajectory. These pathways define virtual places where all the involved roles meet and cooperate, become aware of the activities of each other and articulate their collaboration and usage of resources: in sum, they make sense of the whole actions on the same patient. The process models are the expression of these virtual places and can be used to modulate the conveyance of care-related and patient-related information with the aim to promote collaborative awareness between care-providers, especially in distributed and inter-departmental cooperative settings. Modulation is necessary since the same piece of awareness information should reach different actors with different strength or adapted content in relation to its role, current involvement in the process, availability and the like.

CASMAS manages this modulation by means of a spatial model of awareness [16] in the same line as [2]. Awareness promotion is characterized by a reaction-diffusion mechanism: an entity emits an awareness information that is

propagated in a space according to a predefined diffusion policy and is perceived by another entity according to its current location in that or other spaces and to a sensitivity degree that dynamically depends on its current conditions.

A diffusion policy is the result of the combination of the diffusion function defined for each type of awareness information and the weights assigned to the links/arcs between the sites/nodes of the space, as presented more in detail in Section 3.

When using CASMAS to deal with ICPs, the main idea is *to map each (sub-) process model on a topological space*: each space is a graph of connected sites where each site refers to a single task. In each space, the actors move from site to site according to which specific task they are currently involved in. Since ICPs show an articulated structure of interacting (sub-) processes, their representation in CASMAS may encompass several spaces that constitute a not hierarchical layered structure. What goes on a single space depends on how the related (sub-) process can define specific awareness promotion policies, e.g., who is notified of which information with which strength when traversing a specific site.

Each task is characterized in terms of what are the main documental items that are used as its inputs or filled in at its completion as outputs. This information is modeled by means of another component of CASMAS, i.e. a *common information space* [1] (called *fulcrum* in CASMAS terminology) that can be accessed by all actors involved in the caring processes. Several fulcra can exist since the information they contain can refer to specific groups (i.e., *communities* in CASMAS terminology): however, in this paper, we will consider only one fulcrum since the focus is mainly on the promotion of modulated awareness information.

The fulcra and the multi-layered topologies allow for the provision of contextual warnings, reminders and alerts that are augmented by additional awareness-promoting information (in what follows just *awareness information*). This is performed thanks to the possibility to consider not only data and events that are concurrent with a specific task, but also prospective conditions that are expected in the next steps encompassed by the process model. In this framework, the concept of *distance* expressing how "logically far" two tasks or involved roles are is used in order to limit the problem of information filtering and overload. In fact, the mapping of a process on a topological space allows actors to be characterized also in terms of relative distance from each other, as well as in terms of absolute distance from informative resources, which are connected as inputs and outputs of tasks. This allows for the computational management of the value of awareness information at each site of the topological space that can be perceived by each actor passing through this site.

The multi-layered nature of the model allows for the management of the increasingly complex situations that are quite likely in any multi-disciplinary healthcare setting. For example, the model allows for the contextual generation of information awareness when a set of multiple teams of clinicians have to coordinate with each other along an ICP established for the proper care of a single patient. Or, when multiple teams of practitioners share the care of several

patients, each requiring the compliance to a different and circumscribed care protocol; or when a set of teams of care providers manage the same patient for different illnesses where interventions are not intended to overlap or conflict, but they end by doing it eventually; or when the illness trajectory of a patient requires some team involved in its management to change the intended care protocol, either adopting another protocol or slightly varying the standard one. In all these cases, the possibility to either juxtapose multiple topological layers, to change features characterizing a topology or substitute one process model to another, even on the fly, allows for the data- and event-driven modulation of the propagation of clinical information between actors and sites of the spaces so as to convey the right information to promote collaborative awareness. In this paper, we illustrate the capabilities of CASMAS in some of the cases mentioned above.

3 The CASMAS model

CASMAS is a model conceived to support the design of systems that enable and foster cooperation in a pervasive computing domain. Although CASMAS can deal with devices in the light of pervasive computing (see [5] as an example), in this paper we concentrate on the logic by which proper and timely (awareness) information is generated and we will not consider devices any further. The model derives its name from the heterogeneous multi-agents architecture that defines its operational semantics in terms of mechanisms expressed in terms of rule-based and declarative constructs. Since CASMAS is intrinsically modular, as will be presented in the following, also rules can be defined and organized in a modular and manageable manner. In fact, rules can be defined per entity and per community and can also be aggregated in behaviors (which are sets of rules that when loaded in an entity allow it to enact a specific behavior).

CASMAS is centered on the notion of community whose members share the common goal to make cooperation effective in the given context [5]. Members are modeled as *entities* that stand as proxies for them and incorporate behaviors supporting their activities. Entities can be associated to both *community fulcra* and *community spaces*, where coordination and awareness information is managed, respectively.

A community fulcrum is the place where entities share coordination information: typically, tasks and their relations as well as the information resources they involve, coordinative artifacts and community's conventions [6] that characterize each community.

A community space is where entities perceive awareness information that is propagated in a modulated manner according to a diffusion function. This function makes use of the topology of the space and of the weights that can be associated to arcs for each specific type of information. The diffusion function associates to every node of a space a triple: information type, intensity, and *content*. It has the following parameters: the location of the source and of the

current node and the gradient expressing how the intensity is increased or decreased according to the computed distance. The intensity at a node is related to the form or the strength of how the perceivable content is conveyed by entities reaching this node and owning a sensitivity degree to this kind of information. In this way, different types of information can be propagated at different paces on the same space and then reach different sets of sites accordingly. Community spaces can be dynamically instantiated to respond to the requirements of the current situation, e.g. when a new caring process is enacted and activated.

Different communities can coexist. However, each community is identified by a single fulcrum and is associated to a single set of community spaces. Each fulcrum and each space can be associated only to one community, so as to preserve its identity. However, CASMAS allows that the same entity belongs to different communities, that is, is linked to more than one fulcrum and space. This feature models the migration of information and behaviors across different fulcra so as to support the dynamic joining of entities to the related communities and the adaptation of their behavior according to the information that characterize them (this aspect will not be further described: for more details see [10]).

In addition, CASMAS allows for the migration of awareness information across different spaces: this is done by a mechanism named *interface* that specifies whether a space can export to or import from another space a specific kind of awareness information. Moreover, entities can be dynamically linked to a community space: a typical case is when the latter is dynamically created or when the coordination logic expressed in the fulcrum requires the propagation of specific awareness information toward new entities. All these aspects are illustrated in the following sections through the use of a reference scenario using an articulated ICP.

4 CASMAS at work

Let us consider the following scenario of integrated care around the illness trajectory of an elderly patient.

> An independent elderly person suffering from high blood pressure one day has a stroke attack. She is brought to the Emergency Department (ED), where the triage nurse assesses her functional conditions and suspect a stroke. In the ED the preliminary and general examinations to verify this hypothesis are accomplished. Once admitted to the Stroke Unit (SU), the patient undergoes further examinations and is treated accordingly. After some days, the patient is transferred to the Rehabilitation Unit (RU). From there, after recovering some of the lost functionalities, she is finally discharged home. Such a care plan could be expressed in terms of the ICP depicted in Figure 1: it helps the involved practitioners anticipate needed actions and assessments. It also helps prospective practitioners that *could be* involved in the next phases of the plan be prepared and pre-informed. For instance, within the Emergency

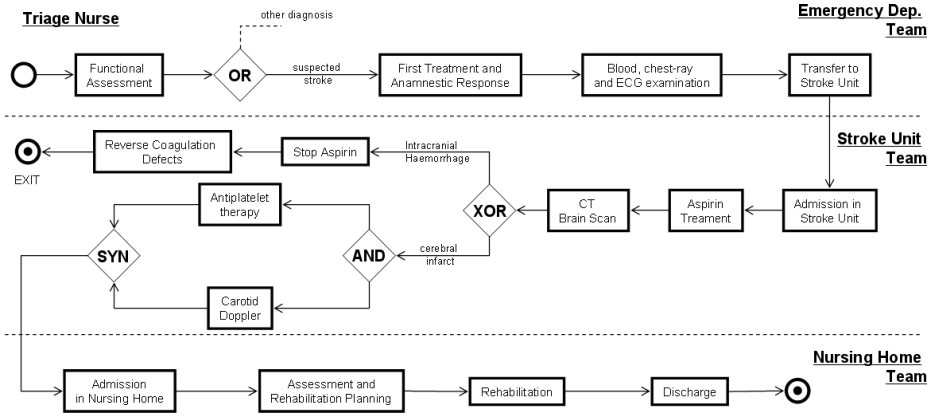

Fig. 1. The ICP adopted and shared by practitioners.

Department, ED nurses could be warned to prepare all the necessary to request an ECG and chest ray no sooner had the triage nurse reported her assessment on the patient record. A different example can take into account the coordinative needs of multiple centers. For instance, practitioners of the Stroke Unit could be pre-informed that a patient would be going to be transferred after that the report of the chest ray (prescribed at the ED) has confirmed the stroke diagnosis.

The ICP presented in Figure 1 is modeled in CASMAS by means of three community spaces (Figure 2), one space for each team involved in the care plan. Each task corresponds to a site of the space (which carries its ID as name); in this manner it is possible to know which task(s) are currently active when entities (corresponding to practitioners) move over the space. Consequently, these entities are able to perform the activities associated to the task by combining community behaviors specified in the fulcrum and EPR's information as described in Section 4.1.

Sites representing tasks that are an "exit" point of a portion of an ICP, such as "Transfer to Stroke Unit", are connected to the "entry" point task of the corresponding portion of this ICP, such as "Admission in Stroke Unit". This connection is realized by means of an *interface* between the two sites; in order to manage the information exchange between the two teams this interface is characterized by the type of information that can flow across them as shown in Section 4.2.

4.1 The features supporting information flow

As anticipated, we consider a single fulcrum (Figure 3) that contains the coordinative information concerning the practitioners community, specifically about the adopted ICP and the inputs, outputs and coordinative behavior of the tasks

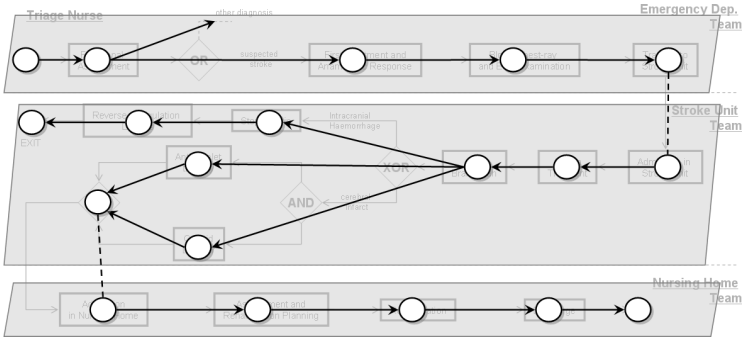

Fig. 2. The CASMAS community space derived from the ICP.

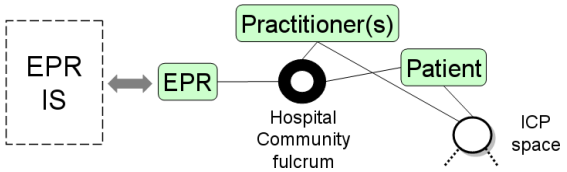

Fig. 3. The integration with the EPR application according to the reference scenario.

constituting it. Since the modeling effort is oriented to the design of applications supporting collaboration, the resulting CASMAS based system has to be integrated with an EPR application with two primary goals: first, from the collaborative application point of view, the EPR provides the user interface for the practitioners and is the source of information about the patients to be used by the CASMAS based system; secondly, from the architecture point of view, the CASMAS based system has to be deployed in the technological environment of the hosting caring organization. To this aim, the EPR application is represented in CASMAS as a proxy entity (Figure 3) that is connected to the hospital community fulcrum in order to share information with the other entities and to communicate with the EPR application to acquire information about the patients and their current care context.

The information flow depends on the location of the various entities in the community spaces. Let us consider the Patient entity. Its movement is indirectly controlled by the practitioners that takes care of the patient. In fact, this movement depends on the next task to be executed. Of course, practitioners are in charge of to decide whether the current task is completed and what is the next task: they provide this information by means of a suitable user interface that, for sake of simplicity, we consider as an extension of the EPR application.

When a practitioner selects the new task, this information goes to the fulcrum so that the Patient entity can react accordingly; on the space where it is situated, the Patient entity *moves* to the site that corresponds to (the ID of) the new task, if the behavior of this task (contained in the fulcrum) involves the Patient entity, and *emits* the "task-status" awareness information. The new

"task-status" information replaces the previously emitted one so that all involved practitioners (especially who did not select the new task) can be aware of the current status of the overall ICP process.

If the next task is an "exit" point of a portion of the ICP, such as "Transfer to Stroke Unit" (Figure 1), then the Patient entity activates a more complex behavior. In fact, the Patient entity has to disconnect from the current space and connect to the site associated to new task in the new space. This is a standard behavior that characterizes the Patient entity as member of the considered community adopting the considered ICP. So, when the patient has to be transferred to the Stroke Unit (which implies that the Patient entity is currently situated at the site C of the "Emergency Dep. Team" space, Figure 4) the Patient entity disconnects itself from the "Emergency Dep. Team" space and connects to the site D (Figure 4) of the "Stroke Unit Team" space. The same mechanism applies when other entities, e.g. Practitioners, derive from their behavior (contained in the fulcrum) the need to move to another site or space: for example, when they are assigned to other duties within the same or another ICP or when they are ending their shift. The above mechanism will support the propagation of awareness information about a patient's allergy, as we present in the following section.

4.2 Awareness propagation and perception

Information promoting awareness on some relevant condition can be propagated through the sequence of next due tasks encompassed within the pathway. Every nodes can modulate this information. Some information can be suppressed, so that its propagation stops in a specific node. Some other information can be either amplified or dampened within a node and so re-transmitted to the next nodes according to the type of the information in the context of the care process.

> Let us suppose that, in the task of gathering anamnestic information (anamnestic response, Figure 1), nurses detect that the patient is allergic to aspirin, a common anticoagulant that can be prescribed in several cases of stroke to prevent tissue infarction. Once they have reported this relevant information on the patient record, an alert information on this allergy could be propagated to several next nodes in the pathway, since it is relevant in most of the therapeutic steps of the care process. Specifically, this information could be propagated up to the node representing the task of anticoagulant administration (see aspirin treatment), a specific activity encompassed by the stroke management pathway when the patient is attended by the Stroke Unit team. Conversely, the information regarding the fact that some task is about to be completed could be propagated just to either the next node or the few next tasks on the agenda of the same team. In doing so, such a task-status information could make practitioners involved in this task aware that they will have to take the patient into their care soon.

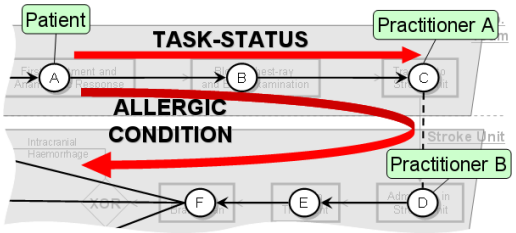

Fig. 4. Propagation of process-based awareness information.

The proposed model supports the modulated propagation of awareness information over the ICP process. The right information flow is ensured by proper weights assigned to the arcs of the space. In fact, a light weight is assigned to each arc according to the process direction (see arrows' direction in Figure 1): conversely a heavy weight is assigned in the opposite direction. In this manner, for example, when awareness information is propagated from the site B (Figure 4) the information propagates towards the site C and not towards the site A.

In addition, the propagation of awareness information is "controlled" also by interfaces between spaces. Each interface is characterized by the type of awareness information that can propagate through it. The interface between the "Emergency Dep. Team" space and the "Stroke Unit Team" is configured to allow only the transit of "allergic" awareness information because, as said in the scenario, this is a relevant information also for the other teams involved in the patient care.

When a practitioner reports in the EPR that the patient is allergic to aspirin, the EPR entity publishes this information in the fulcrum because the EPR entity owns a behavior to make allergy information available by default. The Patient entity owns a behavior to propagate awareness information about allergies. Hence, when the "allergic to aspirin" information is shared in the fulcrum, the Patient entity emits an awareness information from the site of the space where it is situated. Consequently, the allergic information is propagate over the "Emergency Dep. Team" space (from site A to site C, Figure 4), it passes through the interface connected with the "Stroke Unit Team" (from site C to site D), and finally it is propagated over the "Stroke Unit Team" space.

Likewise, when a practitioner selects the next task (as explained in the previous section), the Patient entity emits a "task-status" awareness information from the new task site so that this information is propagated in the space to inform the practitioners involved in the next tasks and let them be prepared to perform it. Modulation allows the designer to identify how far this information has to go and with which strength at each site. Although the "task-status" awareness information might have a positive intensity at the C site (which means that it could be potentially propagated), it is not propagated to the "Stroke Unit Team" because the interface does not allow this type of information to pass through.

Although the propagation of the awareness information is dependent "only" on the process, i.e. it is based on the process structure and on the location of the information sources, practitioners (by means of their proxy entities) can perceive awareness information in different ways. Their perception depends on both the role they play and the process in which they are involved. In fact, each task of the process can contain the information about how single practitioners (playing a specific role) perceive a particular awareness information. In this manner, different roles can differently perceive the same awareness information on the same task site; moreover, a role can be designed to differently perceive the same (type of) awareness information when located at task sites of different processes.

The information about how a role perceives a particular awareness information is stored within the task description in the community fulcrum. An entity of a practitioner that is involved in the patient care process acquires the proper perception behavior by combining the information about the process space where it is situated (which is used to retrieve the process name) and its current position in that space (which is used to retrieve the task ID).

5 Conclusion

A wide literature illustrates how collaborative awareness improves process coordination in distributed work settings, e.g. [15]. The paper has discussed this theme in the light of distributed care supported by process models called Integrated Care Pathways. These can be used as relevant sources of information to propagate this kind of awareness information. The paper has illustrated in a reference scenario how the CASMAS model can support the design of collaborative applications that address awareness promotion: this can be fostered and modulated to reduce information overflow by using specific features of the model. These features mainly concern the possibility to construct flexible mechanisms to modulate awareness propagation according to spaces that capture relevant aspects of the target domain (in this case, ICPs). This possibility is based on a comprehensive model that has been implemented in a rule based environment in order to cope with events whose occurrence can not be strictly anticipated. The mechanisms are highly modular not only because they are defined in terms of rules expressing the reaction-diffusion principle but also because they regard to different not hierarchical and autonomous levels of space definition: in this way, if a space has to be changed, the mechanisms can be reactivated after the change. Finally, the CASMAS model has associated a language that can facilitate the construction of applications built according to it: this aspect has not been presented here for sake of conciseness. However, this is a relevant part of our future work in order to build a framework that better supports the designers in the construction and deployment of collaborative applications.

References

1. Liam Bannon and Susanne Bødker. Constructing Common Information Space. In *Proceedings of the Fifth European Cooperative Supported Cooperative Work*, pages 81–96, Lancaster (UK), 1997. Kluwer Academic Publishers, Netherlands.
2. S. Benford and L. Fahlén. A Spatial Model of Interaction in Large Virtual Environments. pages 109–124, Dordrecht, 1993. Kluwer Academic Publishers.
3. De Bleser, Depreitere, De Waele, Vanhaecht, Vlayen, and Sermeus. Defining pathways. *Journal Of Nursing Management*, 14:553–563, 2006.
4. N. Bricon-Souf, J.M. Renard JM, and R. Beuscart. Dynamic workflow model for complex activity in intensive care unit. *Medinfo*, page 227–31, 1998.
5. Federico Cabitza, Marco Locatelli, and Carla Simone. A community-centered architecture for the deployment of ubiquitous telemedicine systems. In *Healthinf'08: Proceedings of the International Conference on Health Informatics, Funchal, Madeira, Portugal*, volume 1, pages 9–16. IEEE Computer Society, 2008.
6. Federico Cabitza and Carla Simone. "... and do it the usual way": fostering awareness of work conventions in document-mediated collaboration. In *ECSCW'07: Proceedings of the Tenth European Conference on Computer Supported Cooperative Work (ECSCW), Limerick, Ireland*, pages 119–138. Springer, September 2007.
7. G. Fitzpatrick. Integrated care and the working record. *Health Informatics Journal*, 10(4):291–302, 2004.
8. J. Fox, N. Johns, and A. Rahmanzadeh. Disseminating Medical Knowledge - The PROforma Approach. *Artificial Intelligence in Medicine*, pages 157–181, 1998.
9. H. Campbell H, R. Hotchkiss, N. Bradshaw, and M. Porteous. Integrated care pathways. *British Medical Journal*, 316:133–137, 1998.
10. Marco P. Locatelli. *Design of Ubiquitous Collaborative Environments: Supporting Coordination and Awareness in an Integrated Way*. PhD thesis, Università degli Studi di Milano-Bicocca, February 2008. www.itis.disco.unimib.it/research/people/marcoplocatelli/locatellis-phd-thesis.
11. C. Patterson. The integrated care epidemic. *Age and Aging*, 31:157–158, 2002.
12. M. Peleg, A.A. Boxwala, and S. Tu. The intermed approach to sharable computer-interpretable guidelines: a review. *Journal of American Medical Informatics Association*, 11(1):1–10, 2004.
13. Wanda Pratt, Madhu C. Reddy, David W. McDonald, Peter Tarczy-Hornoch, and John H. Gennari. Incorporating ideas from computer-supported cooperative work. *Journal of Biomedical Informatics*, 37(2):128–137, 2004.
14. M. Reddy, P. Dourish, and W. Pratt. Coordinating Heterogeneous Work: Information and Representation in Medical Care. In *ECSCW'01: Proceedings of the European Conference on Computer Supported Cooperative Work, Bonn, Germany*, page 239–258, 2001.
15. Kjeld Schmidt, Christian Heath, and Tom Rodden, editors. *Computer Supported Cooperative Work Journal*, volume 11. Kluwer Academic Publishers, 2002.
16. Carla Simone and Stefania Bandini. Integrating awareness in cooperative applications through the reaction-diffusion metaphor. *Computer Supported Cooperative Work, The Journal of Collaborative Computing*, 11((3-4)):495–530, 2002.
17. Anselm Strauss, Shizuko Fagerhaugh, Barbara Suczek, and Carolyn Wiener. *The Social Organization of Medical Work*. University of Chicago Press., 1985.

Flexibility Schemes for Workflow Management Systems

- regular paper -

R.S. Mans[1], W.M.P. van der Aalst[1], N.C. Russell[1], P.J.M. Bakker[2]

[1] Department of Information Systems, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB, Eindhoven, The Netherlands.
{r.s.mans,w.m.p.v.d.aalst,n.c.russell}@tue.nl

[2] Academic Medical Center, University of Amsterdam, Department of Innovation and Process Management, Amsterdam, The Netherlands.
p.j.bakker@amc.uva.nl

Abstract. Currently, many hospitals are investigating the use of workflow management systems in order to provide support for healthcare processes. However, contemporary workflow management systems fall short in supporting care processes which require flexible execution options. In this paper, we investigate the flexibility requirements that need to be satisfied in order to support various kinds of healthcare processes Our evaluation shows that different systems need to be used in conjunction with each other in order to fully support the various types of care processes.

Key words: workflow management, flexibility, healthcare

1 Introduction

In a competitive health-care market, hospitals need to focus on ways of streamlining their processes in order to deliver high quality and safe care while at the same time reducing costs [7]. Consequently, there is a need for technological support in controlling and monitoring healthcare processes for patients [12] and workflow technology is potentially a means for achieving this end. Workflow Management Systems (WfMSs) support processes by managing the flow of work such that individual work items are done at the right time by the proper person [1]. The benefits being that processes can be executed more rapidly and can be monitored.

A number of difficulties commonly arise when hospitals attempt to automate healthcare processes as a consequence of the fact that these processes are *diverse*, require *flexibility* and that *several medical departments* can be involved in the diagnostic and treatment process. For a group of patients with the same diagnosis, the number of different examinations and treatments required can be high and the order in which they are conducted can vary greatly.

Therefore, an interesting and challenging question arises: *What are the considerations with regard to process flexibility when applying workflow technology*

in hospitals?. When we look at how to achieve process flexibility, four different approaches can be identified [14] which differ in the timing and manner that they are applied. More details can be found in [14].

Flexibility by design: the ability to incorporate alternative execution paths within a process model at design time allowing the selection of the most appropriate execution path to be made at runtime for each process instance.

Flexibility by deviation: the ability for a process instance to deviate at run-time from the execution path prescribed by the original process without altering its process model. The deviation can only encompass changes to the execution sequence of tasks in the process for a specific process instance, it does not allow for changes in the process model or the tasks that it comprises.

Flexibility by underspecification: the ability to execute an incomplete process model at run-time, i.e., one which does not contain sufficient information to allow it to be executed to completion. The model needs to be completed by providing a concrete realization for the undefined parts.

Flexibility by change: the ability to modify a process model at run-time such that one or all of the currently executing process instances are migrated to a new process model.

To answer the previous question, we implemented a *representative* healthcare process in four workflow systems. Based on the above four flexibility types, we will discuss what kind of flexibility is actually needed in order to support the representative healthcare process and healthcare processes in general.

As the representative care process, we have taken the diagnostic process of patients visiting the gynecological oncology outpatient clinic in the AMC hospital, a large academic hospital in the Netherlands. The healthcare process under consideration is a large process consisting of around 325 activities. We choose to implement the care process in workflow systems which demonstrate various kinds of flexibility. For this purpose we selected YAWL [2, 6], FLOWer [5], ADEPT1 [17], and Declare [16]. YAWL was chosen because it is a powerful open-source system supporting most of the workflow patterns [13]. FLOWer is considered to be the most successful commercial system providing flexibility support. ADEPT1 and Declare are two academic systems providing new and powerful ways of supporting "extreme" flexibility. Moreover, the selected systems cover distinct areas of the Process Aware Information Systems (PAIS) technology spectrum, such as adaptive workflow (ADEPT1), case handling (FLOWer), and declarative workflow (Declare). In Table 1, it is shown which flexibility types are supported (+) and not supported (–) by each workflow system. A detailed evaluation can be found in [14]. Together with the identified flexibility requirements this allows for examination of the conditions under which a workflow system can be applied in the healthcare domain. Note that we only focus on the control-flow perspective of a process. Other factors which might be relevant are not considered.

This paper is structured as follows: Section 2 introduces the gynecological oncology healthcare process in general and a subpart of it in detail. Section 3 discusses the corresponding implementation in each of the different workflow systems. Section 4 examines the flexibility needed for supporting healthcare

Flexibility by	ADEPT1	YAWL	FLOWer	Declare
design	+	+	+	+
deviation	–	–	+	+
underspecification	–	+	–	–
change	+	+	–	+

Table 1. Product evaluations.

processes. Related work is outlined in Section 5. Section 6 concludes the paper.

2 Case of gynecological oncology

In this section, we introduce the diagnostic part of the gynecological oncology healthcare process, which we studied. In Figure 1, a snippet showing the most important part of the process is given. Moreover, for the "referral patient and preparations for first visit" node a part of the corresponding subprocess is shown in Figure 2.

Figures 1 and 2 model the gynecological oncology process using so-called *Colored Workflow Nets (CWN)* [4], which are a specific class of *Colored Petri Nets (CPNs)* [10]. Furthermore, a CWN is a *workflow model* in which we restrict ourselves to concepts and entities which are common in most workflow languages.

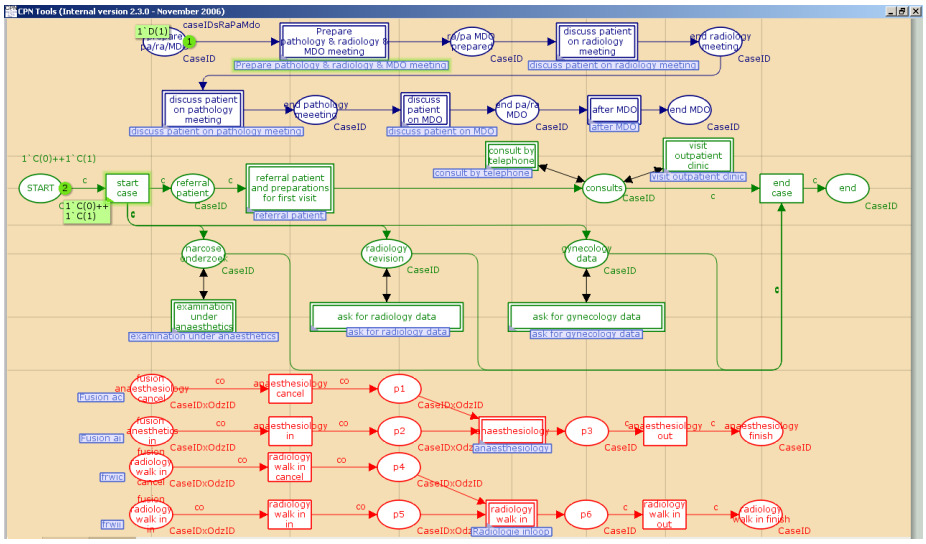

Fig. 1. General overview of the diagnostic process of the gynecological oncology healthcare process. The green and blue nodes and arcs represent respectively the first and second part of the process. The red nodes and arcs represent the interactions with different medical departments.

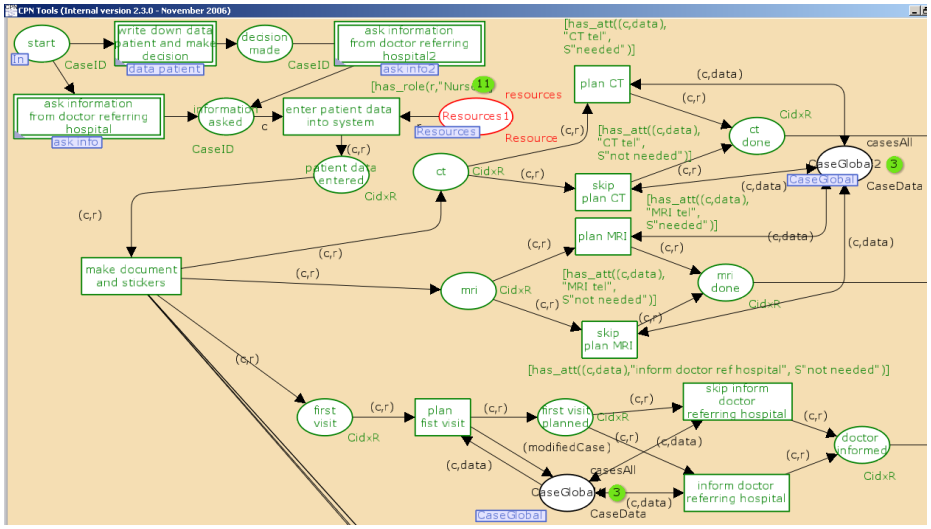

Fig. 2. CWN for the first part of the gynecological oncology healthcare process.

To this end, a CWN covers the *control-flow*, *organizational*, *data* and *operational* perspectives. More details about a CWN can be found in [4]. In Figure 1, the topmost page of the CWN model is shown which gives a general overview of the diagnostic process of the gynecological oncology healthcare process in the AMC hospital. As we are dealing with a large healthcare process it is only possible to show a small part of the *overall* model.

As can be seen in Figure 1, the gynecological oncology process consists of two different processes of which only one will be considered in detail. The process, which is modeled in the lower part of the picture and colored green, deals with the diagnostic process that is followed by a patient when referred to the AMC hospital for treatment, up to the point where the patient is diagnosed. In this process, the patient can have several consultations with a doctor, either by visiting the outpatient clinic or via telephone. During such a consultation, the status of the patient is discussed and a decision is made about whether examinations and consultations need to be requested, canceled, or rescheduled. Moreover, during the course of the process, several administrative activities such as brochure recommendation and patient registration can also occur. A doctor can request various tests, performed by different medical departments. The interactions with these medical departments and also the processes adopted by them are modeled at the bottom of Figure 1 (the red colored nodes). It is important to note, that in the future new tests can become available, even new types of medical departments. In this way, it becomes clear, that in order to cater for varying interactions with medical departments, at runtime we need to decide which interactions are needed which can be provided by flexibility by underspecification.

Having introduced the gynecological oncology process, we will focus on its initial stages (i.e. substitution transition “referral patient and preparation for

first visit"), in which a doctor of a referring hospital calls a nurse or doctor of the AMC hospital resulting in an appointment being made for the first visit of the patient. At that moment it is also necessary to schedule appointments for diagnostic tests. This part of the process is shown in Figure 2. For example, we see that the first visit of the patient needs to be planned, and that it is possible to make an appointment for an "MRI".

The process, shown in Figure 2, is considered to be a "standardized procedure" for these patients at the AMC. From the figure, it can be seen that there a number of possible courses of action that may be taken (and the figure only shows half of the process). Furthermore, as healthcare processes are unpredictable, there can also be the need to skip or to add activities. Respectively, the first is an example of flexibility by design and the latter is an example of flexibility by deviation.

3 Realization of the system in different Workflow Systems

In this section, we will discuss how several different workflow systems have been configured in order to support the healthcare processes discussed earlier. The workflow systems YAWL, FLOWer, ADEPT1 and Declare have been chosen as candidate systems. Each of them demonstrates a specific kind of flexibility, which is deemed relevant when implementing a healthcare process in a workflow context. In the remainder of this section, we will examine how the flexibility provided by each workflow system has been used or can be utilized during the execution of healthcare processes. Due to space limitations, we will do this in detail for YAWL in Section 3.1 and present the main findings for the other systems in Section 3.2.

3.1 YAWL / Worklets

YAWL (Yet Another Workflow Language) [2] is an open source workflow management system, which is based on the well-known workflow patterns [13] and is more expressive than any workflow language available today. YAWL supports the modeling, analysis and enactment of flexible processes through the use of *worklets* [6] which can be seen as a kind of configurable process fragment. Specific activities in a process are linked to a repertoire of possible actions. Based on the properties of the case and other context information, the desired action is chosen. The selection process is based on a set of rules. Also, during enactment it is possible to add new actions to the repertoire.

In YAWL, we used the worklet approach for modeling the interactions with all medical departments by linking a "multiple atomic task" node to the worklet service. This is represented in Figure 3(a) by the node with name "examinations" which can be executed additional times if multiple examinations are needed. In this way, for each test the right worklet can be chosen. In the case where a new test arises, it is possible to choose a corresponding process fragment, or to dynamically define a new process fragment, thereby extending the ruleset.

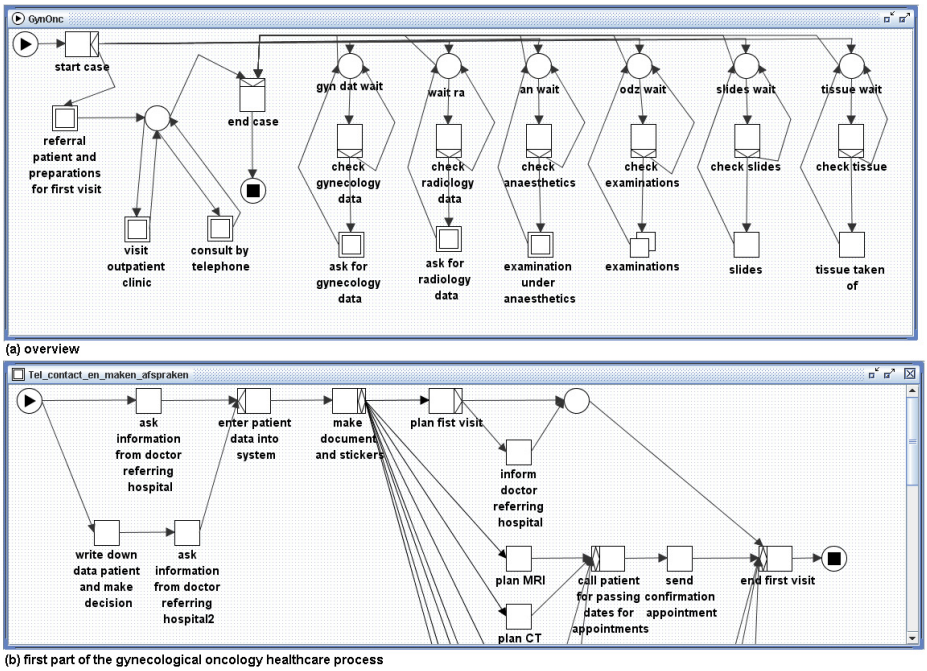

Fig. 3. Screenshots of models in the YAWL editor.

In Figure 3(b), we see the corresponding YAWL process fragment for the first part of the gynecological oncology healthcare process. Due to syntactical sugaring, less nodes were needed than are required in the CWN model of Figure 2. For example, the "make document and stickers" activity in YAWL is an OR-split, which means that one or more of the outgoing paths may be followed and others may be skipped. This OR-split is used because each of the "plan MRI" and "plan CT" activities may or may not be performed.

3.2 Realization in other workflow systems

In this section, the realization in FLOWer, ADEPT1 and Declare is discussed. Each system will be introduced shortly followed by the main findings for the system.

FLOWer *FLOWer* is a commercial workflow management system provided by Pallas Athena in the Netherlands. FLOWer is a case-handling product [5]. Case-handling aids process flexibility by focussing on the data aspect rather than on the control-flow aspect of processes. In particular the following flexibility features offered by FLOWer are used. First, work distribution is separated from authorization, which allows for additional types of actions, like skipping or redoing activities in the process. An example is the skipping of the "plan MRI" step of Figure 2. Second, workers are allowed to view and add/modify data before and

after the corresponding activities have been executed. So, if the activity "make document and stickers" has already been executed, in FLOWer it is still possible to go back in the process to where the activity "enter patient data into system" was executed.

In FLOWer, we used the "dynamic subplan", which allows for concurrent execution of a subprocess, for modeling the interactions with all medical departments. However, if a new test is needed, a new version of the process needs to be introduced. Unfortunately, it is not guaranteed that already running cases can be updated to the new version of the process in a safe and secure way.

ADEPT1 ADEPT1 is an academic prototype workflow system [17], developed at the University of Ulm, Germany. ADEPT1 supports *dynamic change* which means that the process model for one individual case can be adapted. In doing so, it is possible to deviate from the pre-modeled process template (skipping of steps, going back to previous steps, inserting new steps, etc.) in a secure and safe way. That is, the system guarantees that all consistency constraints (e.g., no cycles, no missing input data when a task program will be invoked) which hold prior to the dynamic (ad hoc) modification of the process instance also hold after the modification. The intention of the next version, ADEPT2, is to provide full support for changes, including the propagation of process schema changes to already running instances[8].

So, when realizing the process in Figure 2, in ADEPT1 it is possible, for an already running case, to dynamically add the activity "order drug" after the activity "make document and stickers" and before the activity "plan first visit" which allows for ordering a drug in between the activities "make document and stickers" and "plan first visit".

Declare Declare is another academic prototype workflow system focusing on flexibility [16]. In Declare the language used for specifying processes, called *ConDec*, is a *declarative* process modeling language, which means that it specifies *what* should be done instead of specifying *how* it should be done, as is the case in imperative languages (e.g. YAWL, FLOWer). Users can execute activities in any order and as often as they want, but they are bound by certain specified rules, called constraints. For example, when implementing Figure 2, in Declare we can define that activities "enter patient data into system" and "make document and stickers" needs to be executed at least once, but it is not specified in which order they need to be executed.

Furthermore, Declare also supports *dynamic change*, so that the process associated with individual cases can be adapted. In Declare, this means that it is possible to deviate from the pre-modeled process template by adding or removing activities or constraints. Also, model correctness is guaranteed and it is checked by Declare whether the changes are allowed or not for the cases to which they are applied. As for ADEPT1, we can define that activity "order drug" needs to be done after activity "make document and stickers" and before "plan first visit" by dynamically adding a response constraint between this activity and activity

"make document and stickers" and adding a precedence constraint between this activity and the "plan first visit" activity.

4 Evaluation

In Section 1, four different approaches to achieving process flexibility have been discussed. First, for the case of gynecological oncology we determined which flexibility approach is the best candidate for supporting the healthcare process under consideration. Following on from this, we distinguished different kinds of healthcare processes and offered a basis for classifying their specific flexibility requirements. Finally, we use this classification to evaluate the capabilities of the offerings discussed in Section 3 in order to determine which of them can provide the best support for various kinds of healthcare process.

For the classification we only focus on *organizational* healthcare processes. These processes consist of organizational tasks in which collaboration between people from different departments is a vital process characteristic. Moreover, the process is repetitive, but non-trivial. Unlike medical treatment processes, organizational processes do not provide any support for medical decision making[12]. Note that the focus is on presenting a classification which covers the majority of organizational healthcare processes. It is infeasible to cover everything due to the unexpected character of the processes considered. The classification itself has been made based on the insights obtained when studying the gynaecological oncology healthcare process. Moreover, the classification and the accompanying flexibility requirements are based on discussions with a medical specialist.

Gynecological oncology healthcare process The gynecological process, shown in Figure 1, is performed in an academic hospital (AMC, Amsterdam), and is an organizational process. In general, the art and the number of diagnostic tests to be performed is known. However, the total number of examinations is determined by patient characteristics and previously performed diagnostic tests. Clearly, complex care needs to be delivered in which many different departments can be involved. To this end, *flexibility by underspecification* is an interesting candidate in order to provide support for the process, as it allows for the definition of an incomplete model for which the ultimate realization of tasks can be deferred until runtime.

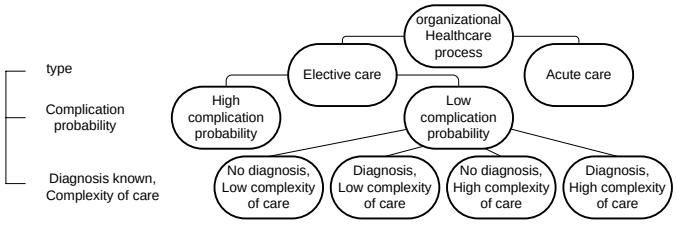

Fig. 4. Classification of healthcare processes.

Healthcare processes In addition to the healthcare process discussed earlier, there exist many other (organizational) healthcare processes with totally different characteristics for which other requirements with regard to flexibility will exist. Figure 4 shows the different types of healthcare processes that can be distinguished. In general, organizational healthcare processes can be divided into:

- *acute care* which deals with critically ill patients in which patient conditions change rapidly; and
- *elective care* for which it is still medically sound to postpone treatment for some days or weeks. Consequently, this kind of care can be planned in advance.

It is clear that acute care cannot be planned and needs to be done in an ad-hoc fashion. To this end, flexibility by change is the best candidate for supporting such an ad-hoc process as the model is not fixed and can be changed into another completely specified model.

Elective care can be planned in advance and several distinct classes of care can be identified. First of all, we propose to make a distinction between processes for which the probability that *complications* arise is *high* or *low*.

Typically, when such a complication occurs it has a high impact on the process as it requires the process to be changed dramatically in some parts. After these changes, the process needs to be made complete again so that it can be executed. Consequently, healthcare processes for which the probability on complications is high can best be supported by flexibility by change.

In contrast, when the complication probability is low, no dramatic changes are to be expected in the process execution. Nevertheless, different classes can be identified which have their own requirements with regard to process flexibility. We propose the following dimensions: *complexity of care* and *diagnosed*. Complexity of care indicates the extent of care which is delivered to a patient, which can be either high or low. Diagnosed indicates whether a diagnosis is known for a patient or not.

In situations where the complexity of care to be delivered is low, more or less, a standard procedure can be followed in which only a few departments are involved. To this end, both the diagnostic and treatment processes can be incorporated in a complete model. Nevertheless, in some cases, occasional unforeseen behavior should be anticipated, where the actual execution at runtime varies from the strict sequence implied by the process model. This can be provided by flexibility by deviation.

However, the complexity of the care to be delivered can also be high. Diagnosing a patient can be very challenging as for some patients it can not be anticipated which diagnostic tests need to be performed. Also, when a patient is finally diagnosed a careful choice needs to be made about the next steps to be done. So, the course of the process is heavily determined by patient characteristics in which collaboration between medical departments is of vital importance. Clearly, for this kind of process, the ultimate realization of some parts of the model needs to be deferred until runtime. This can be provided by flexibility by underspecification.

Flexibility by			design	deviation	under specification	change
Acute care						X
Elective care	High complication probability					X
	Low complication probability	low complexity,no diagnosis	X	X		
		low complexity,diagnosis	X	X		
		high complexity,no diagnosis			X	
		high complexity,diagnosis			X	

Table 2. Flexibility needed for each kind of healthcare process.

Table 2 summarizes which flexibility approach is considered important for which kind of healthcare process. This does not imply that if a flexibility type has not been indicated for a specific type of healthcare process that it is not relevant, rather that it is considered to be of less importance.

System support In Table 1 it can be seen which kind of flexibility is provided by each system. Combining these results with Table 1, we can derive which system(s) can provide the best support for each kind of healthcare process.

The table shows that each flexibility type is relevant for supporting healthcare processes. For both acute care processes and elective care processes with a high complication probability, flexibility by change is needed which can both be provided by ADEPT1 and Declare. For a low complex elective care process, with a low complication probability, a choice needs to be made between flexibility by design and flexibility by deviation. As FLOWer supports both types, this system would be the best candidate. In contrast, for high complex care processes, flexibility by underspecification is needed. To this end, YAWL would be the best candidate.

5 Related Work

Careflow sytems, systems for supporting care processes in hospitals, have special demands with regard to workflow technology. One of these requirements is that flexibility needs to be provided by the workflow system [19]. Unfortunately, current WfMS significantly fall short with regard to providing flexibility, which is seen as a problem in literature [3, 11]. Also, once a workflow-based application has been configured on the basis of an explicit process model, the execution of related process instances tends to be rather inflexible [18]. The workflow systems that we chose in this paper allow for more flexibility than classical workflow systems.

Another requirement when applying workflow technology in the healthcare domain is that real time patient monitoring, detection of adverse events, and adaptive responses to breakdown in normal processes is needed [9]. As adaptive workflow systems are rarely implemented, this makes current workflow implementations inappropriate for healthcare [20]. Furthermore, in a real clinical

setting, it is a critical challenge for any workflow management system that it is able to respond effectively when exceptions occur [15]. Another significant gap that can be identified is that no support is provided for the multidisciplinary nature of healthcare processes. Consequently, there exists the need to support cross-departmental healthcare processes as is stressed in [12].

6 Conclusions

In this paper, we have investigated the flexibility requirements that need to be satisfied by workflow management systems in order to support organizational healthcare processes. As a running example, we used the AMC's gynecological oncology healthcare process which has been implemented in four different workflow systems. For this process, we identified that flexibility by underspecification is a key process requirement, a feature which can best be provided by YAWL.

Furthermore, we identified that different types of healthcare processes each have their own requirements with regard to flexibility. Our results, demonstrate that all flexibility types are useful for supporting specific types of care processes. Individual systems tend to exhibit a degree of specialization in their approach to process flexibility, which has the consequence that different systems need to be used in conjunction with each other in order to fully support all types of care processes that might be encountered. In order to promote the use of workflow management in hospitals, the focus needs to be on enhancing existing tools and/or the development of new ones which provide a greater support for flexibility.

A limitation of our approach is that only one healthcare process has been considered. Future research should focus on implementing healthcare processes with a variety of characteristics in several workflow systems so that deeper insights can be gained into the requirements for process flexibility. In this paper, we only focussed on the control flow perspective of care processes. A further line of research would be to investigate what the flexibility requirements are for other perspectives, such as the data, resource and application perspectives.

References

1. W.M.P. van der Aalst and K.M. van Hee. *Workflow Management: Models, Methods, and Systems*. MIT press, Cambridge, MA, 2002.
2. W.M.P. van der Aalst and A.H.M. ter Hofstede. YAWL: Yet Another Workflow Language. *Information Systems*, 30(4):245–275, 2005.
3. W.M.P. van der Aalst and S. Jablonski. Dealing with Workflow Change: Identification of Issues and Solutions. *International Journal of Computer Systems, Science, and Engineering*, 15(5):267–276, 2000.
4. W.M.P. van der Aalst, J.B. Jørgensen, and K.B. Lassen. Let's Go All the Way: From Requirements via Colored Workflow Nets to a BPEL Implementation of a New Bank System. In R. Meersman and Z. Tari et al., editors,

CoopIS/DOA/ODBASE 2005, volume 3760 of *Lecture Notes in Computer Science*, pages 22–39. Springer-Verlag, Berlin, 2005.
5. W.M.P. van der Aalst, M. Weske, and D. Grünbauer. Case Handling: A New Paradigm for Business Process Support. *Data and Knowledge Engineering*, 53(2):129–162, 2005.
6. M. Adams, A.H.M. ter Hofstede, D. Edmond, and W.M.P. van der Aalst. Dynamic. extensible and context-aware exception handling for workflows. In F. Leymann F. Curbera and M. Weske, editors, *Proceedings of the OTM Conference on Cooperative Information Systems (CoopIS 2007)*, volume 4803 of *Lecture Notes in Computer Science*, pages 95–112. Springer-Verlag, Berlin, 2007.
7. K. Anyanwu, A. Sheth, J. Cardoso, J. Miller, and K. Kochut. Healthcare Enterprise Process Development and Integration. *Journal of Research and Practice in Information Technology*, 35(2):83–98, May 2003.
8. P. Dadam, M. Reichert, S. Rinderle, M. Jurisch, H. Acker, K. Göser, U. Kreher, and M. Lauer. Towards truly flexible and adaptive process-aware information systems. In R. Kaschek, C. Kop, C. Steinberger, and G. Fliedl, editors, *UNISCON*, volume 5 of *Lecture Notes in Business Information Processing*, pages 72–83. Springer, 2008.
9. Y. Han, A. Sheth, and C. Bussler. A taxonomy of adaptive workflow management. In *Proceedings on CSCW-98 Workshop Towards Adaptive Workflow System*, 1998.
10. K. Jensen, L.M. Kristensen, and L. Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer*, 9(3-4):213–254, 2007.
11. M. Klein, C. Dellarocas, and A. Bernstein, editors. *Adaptive Workflow Systems*, Special Issue of Computer Supported Cooperative Work, 2000.
12. R. Lenz and M. Reichert. IT Support for Healthcare Processes - Premises, Challenges, Perspectives. *Data and Knowledge Engineering*, 61:49–58, 2007.
13. N. Russell, A.H.M. ter Hofstede, W.M.P. van der Aalst, and N. Mulyar. Workflow Control-Flow Patterns: A Revised View. BPM Center Report BPM-06-29, BPMcenter.org, 2006.
14. N.A. Mulyar, M.H. Schonenberg, R.S. Mans, N.C. Russell and W.M.P. van der Aalst. Towards a Taxonomy of Process Flexibility (Extended Version). BPM Center Report BPM-07-11, BPMcenter.org, 2007.
15. S. Panzarasa and M. Stefanelli. Workflow management systems for guideline implementation. *Neurological Sciences*, 27:245–249, June 2006.
16. M. Pesic, M.H. Schonenberg, N. Sidorova, and W.M.P. van der Aalst. Constraint-based workflow models: Change made easy. In Robert Meersman and Zahir Tari, editors, *OTM Conferences*, volume 4803 of *Lecture Notes in Computer Science*, pages 77–94. Springer, 2007.
17. M. Reichert, S. Rinderle, and P. Dadam. ADEPT Workflow Management System. In W.M.P. van der Aalst, A.H.M. ter Hofstede, and M. Weske, editors, *BPM 2003, Proceedings*, volume 2678 of *Lecture Notes in Computer Science*, pages 370–379. Springer, 2003.
18. S. Sadiq, O. Marjanovic, and M.E. Orlowska. Managing Change and Time in Dynamic Workflow Processes. *International Journal of Cooperative Information Systems*, 9(1-2):93–116, 2000.
19. M. Stefanelli. Knowledge and Process Management in Health Care Organizations. *Methods Inf Med*, 43:525–535, 2004.
20. J. Sutherland and W.-J. van den Heuvel. Towards an Intelligent Hospital Environment: Adaptive Workflow in the OR of the Future. In *Proceedings of the 39th Hawaii International Conference on System Sciences*, 2006.

From Paper Based Clinical Practice Guidelines to Declarative Workflow Management

Karen Marie Lyng[1] Thomas Hildebrandt[1] and Raghava Rao Mukkamala[1]

[1] IT university of Copenhagen, Rued Langgaardsvej 7, 2300 Copenhagen S, Denmark
{lyng, hilde, rao}@itu.dk,

Abstract. We present a field study of oncology workflow, involving doctors, nurses and pharmacists at Danish hospitals and discuss the obstacles, enablers and challenges for the use of computer based clinical practice guidelines. Related to the CIGDec approach of Pesic and van der Aalst we then describe how a sub workflow can be described in a declarative workflow management system: the Resultmaker Online Consultant (ROC). The example demonstrates that declarative primitives allow to naturally extend the paper based flowchart to an executable model without introducing a complex cyclic control flow graph.

Keywords: Process modelling in healthcare, Process oriented system architectures in healthcare, IT support for guideline implementation and decision support, Requirements for medical guideline and medical pathway support, integrating healthcare processes with electronic medical records.

1 Introduction

It has been known for quite a while that there is a need for making clinical working practices safer, as too many errors happen causing suffering or even death of patients [1]. Due to the complexity, the high mobility and ephemerality of the daily clinical work [2,3] safer working practises will require better coordination, efficient collaboration and not least fulfilment of up to date clinical practice guidelines (CPG) [4-6]. One way of supporting this is by the use of of IT based clinical decision support and better linkages in and among IT-systems [7]. Indeed, according to [8,9] on of the best options for improvement in clinical work seems to be IT supported clinical processes based on CPG's.

However, the use of IT based CPG's is challenging in several ways. Firstly, due to continuous development of new knowledge within the medical domain the mean survival time of clinical guidelines is short, approximately 2 years [10]. Secondly, there is a need for guidelines to be flexible and adaptable to the individual patient [11].Thirdly, no coherent theoretical framework of health professional and organizational behaviour and behaviour change has yet been established [12]. Finally, it is a serious challenge that health professionals currently tend *not* to follow clinical guidelines [5]. One of the reasons for this could be that clinical guidelines are not

embedded in the clinical work processes and the technology available in the clinical setting today.
Oncology is an example of a clinical speciality for which it is known that there do exists a high number of CPGs that are followed to a certain degree by the health professionals [13]. For this reason we found it of interest to perform a series of field studies in oncology clinics, to examine enablers and obstacles for use of IT-supported clinical guidelines. The field studies are presented in Section 2 below. Based on the field studies, we then proceeded in Section 3 to investigate how the current paper based workflows could be supported using a commercial declarative workflow management system, which relates to the CIGDec approach of Pesic and van der Aalst [14]. We believe that the resulting model rather naturally extends the paper based flowchart table used at the hospitals, and in particular avoids the introduction of complex cyclic control flow graphs and over specification as also pointed out in [14]

2 Field study - usage of CPGs in Danish oncology clinics

2.1 Method

Observations were made on three Danish oncology clinics by two observers (the first author and an assistant). Four days of observation were made at each clinic. Besides observations, access to all clinical guidance material was granted. All the clinics were specialized within oncology; two of them were university clinics. The focus of the observation study was on the use of CPG's as defined by Field and Lohr[14]: *Clinical practice guidelines are systematically developed statements to assist practitioner decisions about appropriate health actions for specific clinical circumstance*. We especially looked at the work of nurses, doctors and pharmacists in relation to chemotherapeutical treatment of patients.

2.2 Overall treatment processes and guidance documents

Patients are referred to the clinics with a diagnosis of cancer. By the first visit in the outpatient clinic the patient is informed about pros and cons of chemotherapy by a doctor, and an overall patient plan for oncological treatment is outlined. In subsequent visits chemotherapy is given, in between visits to the outpatient clinic monitoring of side effects to chemotherapy are done by laboratory tests.

The chemotherapeutic treatment is based on a number of different types of guidance documents and diagrams depicted in Figure 1. The basis of the treatment is given in a *standard treatment protocol* or a *research protocol*, which constitute the CPG for the diagnosis in case. The protocols are written in a narrative form with a description of the current knowledge of treatment of the diagnosis as well as a thorough description of the drugs to be used. The size of a research protocol is app. 60-80 pages and a standard treatment protocol is app. 30-40 pages. Protocols are generally developed in cooperation between several oncology departments, frequently

with a pharmaceutical company as a main sponsor and actor. Research protocols are often multinational.

Based on the protocols *local practice guidelines* (also referred to as standard treatment plans) are made as well as a treatment overview, in daily speech referred to as the "noughts and crosses" diagram. The noughts and crosses diagram describes the planned medical treatment as well as examinations during the patient trajectory. There will often be deviations from the original plan due to side effects to treatment, other medical problems or resource problems in the hospital. Each department has some 60-80 standard treatment plans in use plus several hundreds general CPG's, some of which are specific to the oncology department, e.g. treatment of side effects to chemotherapy and some that are shared by several departments e.g. treatment of diabetes.

The flow of each chemotherapeutic treatment session is guided by the so-called patient *flowchart*, which also records the state of the treatment session. Below we will describe the workflow resulting from the flowchart in more detail; this will be the focus of the remaining part of the paper.

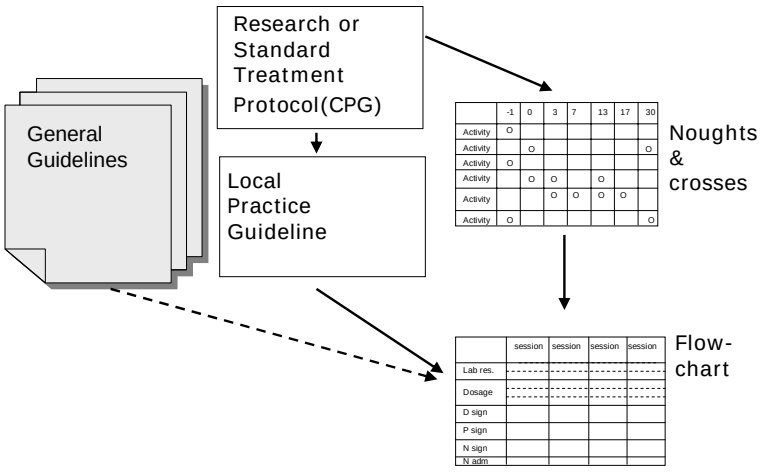

Figure 1 overview of the relation between research protocols/standard treatment plans, local practice guidelines (standard plans) and flow charts. General guidelines are the general guidelines in use at the hospital, containing issues like the treatment of diabetes.

2.3 Current workflow for chemotherapy treatment sessions.

Figure 2 shows an overview of the workflow, which is reiterated in every chemotherapeutic treatment session. In the flowchart the basic information about the patient is registered, including the latest lab results as well as height and weight of the patient. Based on these informations and the patient history of any major adverse effects, the doctor calculates the therapeutic doses of chemotherapy, documents it on

the flowchart and signs it. The flowchart is transferred from the doctor to the controlling pharmacist (who can be situated near by in the clinic or far away in the pharmacy) where it functions as a prescription from the doctor. The controlling pharmacist controls the doctors dosage calculation and writes the information in a working slip that is used for the pharmacy assistant who is doing the preparation of the drug(s) in case. During preparation the quantity of all products as well as batch numbers are registered in the working slip, finally the working slip is signed by the pharmacy assistant, and the product - usually a drip bottle or a pump with a content and patient information label stuck to it – is referred to the controlling pharmacist for check out. When the controlling pharmacist has checked that the produced drug and patient information label matches the flowchart and the working slip, the pharmacist put small green ticks on each item in the flowchart and finally signs it. Subsequently the flowchart and the product is referred to the treatment rooms, where the responsible nurse together with another authorised person (nurse or doctor) checks that the product and flowchart matches, both regarding content and patient information. The responsible nurse then signs the flowchart and the medicine is administered to the patient. In parallel to this the nurse will administer adjuvant medicine like anti-emetics, cortisol and other drugs that are prescribed in the local practice guidelines. The nurse registers the medication in the Medicine Order and Administration (MOA) IT system that currently is being implemented in all the oncology departments.

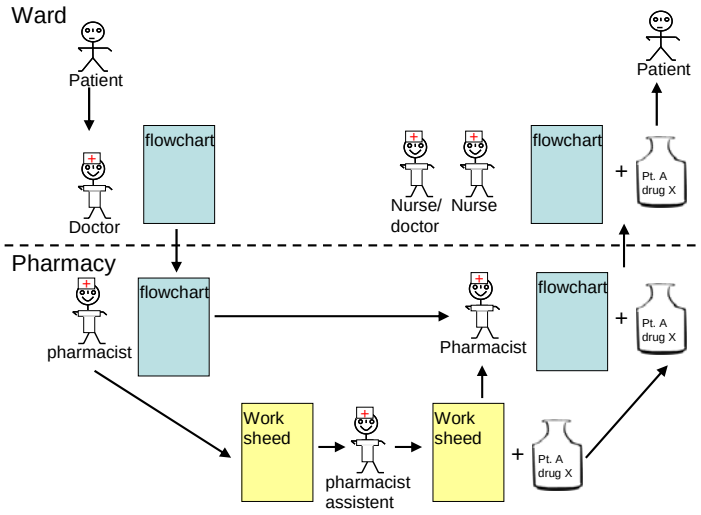

Figure 2 Oncologic workflow in relation to chemotherapeutic treatment of patient

2.4 Preliminary conclusion to the case study

Several characteristics of the work were elucidated in the case study:

- There are several professional actors involved in even rather simple workflows like the ones we studied (they are all involved in more than one workflow at the same time).
- The flow is guided by the flowchart, which is simply a table with a column to which the doctor and pharmacist add information and/or a signature, thereby capturing the state of the session.
- The workflow is distributed: the doctor and nurse, pharmacist, and pharmacy assistant are physically located in different places at the hospital and the current paper used for controlling the workflow is physically transferred by a porter or nurse (or faxed) between the different actors.
- Only the actor currently possessing the flowchart knows its state. Much time was used waiting for and controlling the status of the former process step, to be able to plan own work.
- There are a number of check-points. If a check fails (e.g. the pharmacist or nurse doubts the validity of the current state, the previous actors are asked to verify the state and possibly redo a calculation.
- Exceptional events like the medicine getting too old (e.g. if it is not transferred to the treatment rooms and approved within 24 hours) also led to recurrence of activities.
- Only the state (information) and the actors are explicit in the flowchart. The ordering of events (i.e. transfer of the flow chart between actors), handling of exceptions and recurrence/validation of calculations are implicit.

In our observations we found several potential enablers and obstacles to digitalization of the process support, which have been collected in Figure 3 below.

Enablers	Obstacles
• *Easy access to workflow status, could avoid a lot of walking between treatment rooms and pharmaceutical preparation rooms for the nurses.*	• *Feeling of competence. "I have been here for a hundred years, so I know what to do, and I know the procedures" – guidance are not sought for.*
• *Many patients had to follow more than one CPG, due to co-morbidity or adverse effects of treatment. An It system could present concurrent CPG's*	• *Oral culture – problems are preferably discussed with peers, even rather fact based ones*
• *Meeting legal demands: In the current situation, the pharmacist is lacking a copy of the prescription, which is a legal demand. This could be saved automatic using IT.*	• *No clinical managerial pressure – It is not expected that professionals look things up in the existing sources (Paper or IT-based). There is no control (no count on hits)*
• *It was clear from our observations that CPG's and standard treatment plans was more vividly used if they were embedded in the work processes. This could be in the form of documentation templates, automated order forms or decision algorithms.*	• *Reluctancy to change from paper based workflows*
• *New-commers are more active users of CPG's. In departments with a high turn around of employees process support will be more sought for.*	• *Lack of integration between process support and all the clinical information systems, among which some are still not digitalised.*
• *Experience among clinicians that relevant guidelines are hard to find in current systems*	• *Lack of access to computers, with low response time and single sign on to (all) the clinical IT-systems*

Figure 3 Enablers and obstacles for digitalized clinical process support

In the present paper we concentrate on how the workflow of a single chemotherapeutic treatment session may be supported by a workflow management system, and in particular how the workflow can be described as an executable process. A central issue is how to make the implicit ordering of events (and the additional verifications and possibly recurrences of events) explicit. One option is to use an *imperative* flow graph based notation such as Petri Net or BPMN. However, it would include arrows for capturing the control flow (including cycles for the verification and recurrence of events), which would differ radically from the notation used in the current paper based setting. As suggested by van der Aalst and Pesic in [15] one can avoid introducing the explicit control flow as a complex flow graph by instead using a *declarative* notation such as the CIGDec model. Following this idea, we will investigate below how to specify the treatment session in a commercial declarative workflow management system, the Resultmaker Online Consultant.

3 Resultmaker Online Consultant Model of Treatment Workflow

The Resultmaker Online Consultant (ROC) is a user-centric declarative workflow management system based on a shared data store. It uses so-called *eForms* as its principal activities and allows one to declare the sequential constraints and dynamically included verification steps (and implied recurrences of activities) as found in the oncology treatment workflow using so-called *sequential* and *logical* predecessor constraints and a notion of *activity conditions*.

There is yet no formal graphical notation for the ROC processes, but there is a guideline for how consultants jointly with domain experts can identify and specify activities, roles/actors and constraints in a table of a specific form. This table is referred to as the *Process Matrix* (PM). In Table 1 below shows an example of a PM (simplified to preserve space) for the Oncology workflow presented in the previous section. Each row of the matrix represents an activity of the Oncology workflow. The columns are separated in 3 parts: The first set of columns describes the access rights for the different roles: Doctor (D), Nurse-I (N1), Nurse-II (N2), Controlling Pharmacist (CP), Pharmacist assistant (PA). The next set of columns describes (sequential and logical) predecessor constraints. The last set of columns describes activity conditions. Below we describe the PM for the Oncology workflow and the primitives of the ROC in more detail.

Activities and execution. The notion of an activity in ROC is like in any other workflow language, which means an activity is atomic and corresponds to a logical unit of work. Activities are executed in parallel by default and they can be executed any number of times, unless constrained as described below. The state of the ROC records whether an activity has been executed or not. If an activity has been executed, then that activity will have status *executed*. Its state can be reset under certain circumstances explained in Control Flow Primitives sub section. We say that the flow

has state *complete* at any point where all activities (currently included in the flow, see Activity Conditions below) have state executed.

S No	Activities	Roles					Prede-cessors		Activity Condition	Remarks on data and activity condition
		D	N1	N2	CP	PA	Seq	Log		
1.1	**BASIC_INFO**									
1.1.1	Basic info registration*	W	W	R	R	N				patient information like height, weight and surface area
1.1.2	lab. Results *	W	W	R	R	N				Check lab results
1.1.3	Patient history*	W	R	R	R	N				Interview of patient
1.2	**ORDINATION**						1.1			1.2.2 digitally signs data of 1.2.1 and sets TrustO true. 1.2.3 either sets TrustO true or resets 1.2.1
1.2.1	Calculate the therapeutic doses of chemo-therapy*	W	R	R	R	N				
1.2.2	Sign	W	R	R	R	N		1.2.1		
1.2.3	Verify ordination	W	R	R	R	N	1.2.2		Not TrustO	
1.3	**CONTROL**									
1.3.1	Control calculation	R	R	R	W	R		1.2.2		Set TrustO false if ordination not trusted
1.4	**PREPARE**									
1.4.1	Quantity and batch nr of products are registered*	D	D	D	R	W		1.3.1		This is internal pharmacy work
1.4.2	Sign	R	R	R	R	W		1.4.1		
1.4.3	Check out drip bottle	R	R	R	W	R		1.4.2		1.4.3 resets 1.4.1 if preparation does not match ordination & patient. 1.4.5 resets 1.3.1 or sets TrustP
1.4.4	Sign	R	R	R	W			1.4.3		
1.4.5	Verify preparation	R	R	R	W		1.4.4		Not TrustP	
1.5	**MEDICIN ADM.**							1.4		
1.5.1	Check that preparation, order and patient match	R	W	R						The responsible nurse checks together with another nurse or doctor. If it is not trusted either TrustO or TrustP is set to false (forcing the doctor or pharmacist to verify)
1.5.2	Check that preparation, order and patient match	W	R	W						
1.5.3	Sign	R	W	R				1.5.1 1.5.2		
1.5.4	Admin preparation to patient*	R	W	W				1.5.3		

Table 1 Information marked with * could be transferred from or registered automatically in another hospital information system (HIS) W= write, R = read, N = denied access

The following pre-defined activity types are used in our case:

eForm Activity: *eForms* are web questionnaires that have graphical user interface elements displayable in a web browser. The fields on the *eForms* are mapped to variables in the shared data store and the data filled in by the users will be available to all activities of the workflow instance. *eForms* are appended to ROC activities in process definitions and at run-time when an *eForm* activity is executed, the corresponding *eForm* will be displayed to the user for human interaction. All activities in the example, except signing activities, are eForm activities.

Signing Activity: The user data on *eForms* will be digitally signed by using XML digital signatures syntax and user's digital identity certificates. A single signing activity supports signing of data from multiple *eForms*. In the example all the activities named Sign are signing activities.

Resources/Roles. The ROC supports a simple resource model using Role-based access rights to define permissions on the activities to different users of the system. The possible access rights are Read (R), Write (W), Denied (N) and the default access right on activities is Read access. The Read access right allows a user with the particular role to see the data of an activity, where as Write access right allows the user to execute an activity and also to input and submit data for that activity. A Denied access right is the same as making an activity invisible to the user, i.e. the user does not see it as part of the flow. In the example we have used the denied access right to shield the Pharmacist assistant from the rest of the workflow.

Control Flow Primitives. The control flow primitives define the constraints that control the activity execution at runtime.

Activity Condition: Every activity in the ROC has a logical activity condition. An activity condition is a Boolean expression that can reference the variables from the shared data store. If an activity condition is evaluated to be true, the activity is included in the workflow, otherwise the activity will be skipped. Activity Conditions in ROC workflow model are re-evaluated whenever necessary, so the inclusion of an activity can be changed during the lifetime of the workflow instance. If the activity condition changes to false during the execution of an activity (e.g. when a user is filling in an eForm), the user will be informed that the activity is no longer part of the flow and no data will be changed. This guarantees atomicity of activities. In the example we use two Boolean variables TrustO and TrustP to control the inclusion of the verification actions 1.2.3 and 1.4.5 respectively. When the doctor signs the ordination in activity 1.2.2, TrustO is also set to false, thereby excluding the verification from the flow. However, it may be set to true during activity 1.3.1, 1.51 or 1.5.2. This will force the verification step to be executed and all activities having it as logical predecessor to be reset (see below).

Sequential Predecessors: If activity A is declared to be a sequential predecessor of activity B, then activity B can only be executed if activity A has state "executed". However, the sequential predecessor *has only effect* if the predecessor activity A is included in the workflow instance: This means, that if the activity condition of activity A at a given point of time is false, then the execution of B will not depend on

whether the state of activity A is executed or reset. Sequential predecessor constraints are marked in the Predecessor (Seq) column in the example. For instance, Activity 1.2.2 is a sequential predecessor of activity 1.2.3 (Verify), capturing that it does not make sense to verify an ordination if it has not been signed. Also, every activity in the group 1.1 is sequential predecessors of every activity in group 1.2

Logical Predecessors: If activity A is declared to be a logical predecessor of activity B, then activity A is a sequential predecessor of activity B with additional constraints: Whenever activity A is re-executed, then activity B is reset. Also, if the state of activity A is reset (as described below), then activity B cannot execute again until activity A has been executed again. Like for the sequential predecessor, the logical predecessor constraint between activities A and B has only effect at the point of times where activity A is part of the workflow instance. However, if a logical predecessor activity A becomes part of the workflow instance after activity B has been executed due to the state changes, then the state of activity B will be reset and hence the activity B must be executed once again. In the example, the verification action 1.2.3 may reset activity 1.2.1 (if the doctor finds out during verification that he needs to recalculate the ordination). This again causes activity 1.2.2 to be reset, since it has activity 1.2.1 as a logical predecessor.
To allow for more fine-grained constraints, the ROC workflow model also includes an additional advanced feature called dependency expressions. Dependency expressions are a set of expressions attached to an activity. Like activity conditions, dependency expressions can also contain references to variables from the shared data store. However, an Activity Condition evaluates to Boolean values, dependency expression can evaluate to any value. Any change in the value of the dependency expression will change the activity status to *reset* to indicate that the activity must be executed (at least) one more time (unless it is excluded by the workflow). We have not used dependency expressions in our example.

4 Discussion

It is well known that healthcare processes are complex [17] and although much time is used on coordination [18] errors happens too frequently [1]. CPG's can support healthcare employees in the process of following best practice consistently [6,19], but it is also well known that impediments to access relevant guidelines is an obstacle for use [20] [21] Thus it seems obvious to embed CPG's in clinical IT- process support, although the success of such projects has not yet been convincing [9,22].

In our case study of a rather simple clinical work process we found that the process had an extension in both time and location and several actors was included. Although the process was frequently repeated there were also frequent alterations and recurrences due to returns to previous steps in the workflow. These challenges could be supported in a natural way by the declarative primitives in the ROC workflow management system. As well as the activity conditions allow smooth combination of several sub-workflows. This could be a way of implementing the "noughts & crosses" diagram. Though one have to be aware that IT based business support will lay the

grounds for new work processes, so one should not just automate existing paper based work processes [23]. It was also clear from our observation study that the descriptions in a CPG and the daily work practice are not congruent, this discongruence have to be managed while designing and implementing work process support.

Health professionals are a heterogeneous group, some with little and some with immense experience within a field. Although experience may not totally protect a clinician from committing errors the risk is less and the source of annoyance from detailed guidance by the IT system will be huge. In the ROC focus is on the overall clinical managerial process, for the inexperienced there are links to CPG's outside the ROC. Nevertheless it will be a cultural challenge for clinicians to have a clinical process system directing the road ahead [24], as well as it will have impact on the training and socialisation of new comers to the field [25].

The communication culture in the healthcare sector is profoundly oral [26]. We observed several examples of clinicians discussing factual topics to which the reply only would be a few clicks away. The cultural element will always be a challenge when implementing new technology, especially when it fundamentally changes the work processes [27]

5 Conclusion and Future Work

We believe that IT based process support has a potential in relation to chemotherapeutic treatment of cancer patients. It is though important to be aware that such a change in the clinical work is not just a question of giving access to the right applications. Access to the right equipment as well as integrations of it-systems is mandatory. Also the organisational workflows have to be analysed further and maybe changed [28]. This demands managerial support. More work has to be done to understand the work processes in healthcare and the organisational and social implications of introducing IT support. To obtain knowledge about organisational and social implications it is important to establish carefully planned experiments with process support in clinical settings.

The restricted use of IT in the places we visited can be due to several reasons, it was although clear that the current IT support was incoherent and did not support the clinical way of working. A more thorough unravelling of the clinical processes and the need for information or opportunity to document is a precondition for succeeding with process support [28]. Even a rather simple workflow as the one we have examined unveiled the need for a business process support application to be integrated to several other of the hospital information systems [9,28]. Such an integration provides several challenges, both in relation to access control [29] and in relation to semantics [30,31].

The mapping of the chemotherapy workflow into the Resultmaker Online Consultant demonstrates the use of a commercial workflow model based on declarative process primitives as advocated by Pekic and van der Aalst. The resulting model rather naturally extends the paper based flowchart table used currently at the hospitals, in particular one avoids introduction of cyclic graphs. As future work we plan to present the actors at the hospitals for the ROC model and compare it to other approaches, in

particular the CIGDec language and imperative languages such as BPMN [32]. We also plan to experiment with prototypes of pervasive user interfaces to the ROC.

References

1. Kohn LT, Corrigan JM, Donaldson MS: To Err is Human. Building a Safer Health System. Washington DC, National Academic Press, 2000.
2. Bardram JE, Bossen C: Mobility Work: The Spatial Dimension of Collaboration at a Hospital. Comput. Supported Coop. Work 2005;14:131-160.
3. Bødker S, Christiansen E: Designing for ephemerality and prototypicality. In Proceedings of the 5th conference on Designing interactive systems: processes, practices, methods, and techniques. Cambridge, MA, USA, ACM, 2004.
4. Davis DA, Taylor-Vaisey A: Translating guidelines into practice. A systematic review of theoretic concepts, practical experience and research evidence in the adoption of clinical practice guidelines. Can Med Assoc J 1997;157:408-416.
5. Cabana MD, Rand CS, Powe NR, Wu AW, Wilson MH, Abboud PA, Rubin HR: Why don't physicians follow clinical practice guidelines? A framework for improvement. Jama 1999;282:1458-1465.
6. Grol R, Grimshaw J: From best evidence to best practice: effective implementation of change in patients' care. The Lancet 2003;362:1225-1230.
7. Bates DW, Cohen M, Leape LL, Overhage JM, Shabot MM, Sheridan T: Reducing the frequency of errors in medicine using information technology. J Am Med Inform Assoc 2001;8:299-308.
8. Mulyar N, Pesic M, van der Aalst WM, Peleg M: Towards the Flexibility in Clinical Guideline Modelling Languages. 2008.
9. Lenz R, Reichert M: IT support for healthcare processes - premises, challenges, perspectives. Data Knowl. Eng. 2007;61:39-58.
10. Shojania KG, Sampson M, Ansari MT, Ji J, Doucette S, Moher D: How quickly do systematic reviews go out of date? A survival analysis. Ann Intern Med 2007;147:224-233.
11. Quaglini S, Stefanelli M, Lanzola G, Caporusso V, Panzarasa S: Flexible guideline-based patient careflow systems. Artif Intell Med 2001;22:65-80.
12. Grimshaw JM, Thomas RE, MacLennan G, Fraser C, Ramsay CR, Vale L, Whitty P, Eccles MP, Matowe L, Shirran L, Wensing M, Dijkstra R, Donaldson C: Effectiveness and efficiency of guideline dissemination and implementation strategies. Health Technol Assess 2004;8:iii-iv, 1-72.
13. Goffin JR, Savage C, Tu D, Shepherd L, Whelan TJ, Olivotto IA: The difference between study recommendations, stated policy, and actual practice in a clinical trial. Ann Oncol 2004;15:1267-1273.
14. Field MJ, Lohr KN: Guidelines for Clinical Practice: From Development to Use. Washington DC, Institute of Medicine, 1992.
15. van der Aalst W, Pesic M: DecSerFlow: Towards a Truly Declarative Service Flow Language in Web Services and Formal Methods. Lecture

Notes in Computer Science. Springer Berlin / Heidelberg, 2006, vol 4184/2006, 1-23.
17. Drucker PF: The New Realities. New York, Harper & Row, 1993.
18. Reddy MC, Dourish P, Pratt W: Coordinating heterogeneous work: information and representation in medical care. In Proceedings of the seventh conference on European Conference on Computer Supported Cooperative Work. Bonn, Germany, Kluwer Academic Publishers, 2001.
19. Sim I, Gorman P, Greenes RA, Haynes RB, Kaplan B, Lehmann H, Tang PC: Clinical decision support systems for the practice of evidence-based medicine. J Am Med Inform Assoc 2001;8:527-534.
20. Thorsen T, Mäkelä Me: Changing Professional Practice. DSI - Danish Institute for Health Services Research and Development, 1999.
21. Feder G, Eccles M, Grol R, Griffiths C, Grimshaw J: Clinical guidelines: using clinical guidelines. Bmj 1999;318:728-730.
22. Ash JS, Berg M, Coiera E: Some unintended consequences of information technology in health care: the nature of patient care information system-related errors. J Am Med Inform Assoc 2004;11:104-112.
23. Berg M, Toussaint P: The mantra of modeling and the forgotten powers of paper: A sociotechnical view on the development of process-oriented ICT in health care. Int J Med Inform 2003;69:223-234.
24. Berg M, Horstman K, Plass S, van Heusden M: Guidelines, professionals and the production of objectivity: standardisation and the professionalism of insurance medicine. Sociology of Health & Illness 2000;22:765-791.
25. Mimnagh C, Murphy M: Junior doctors working patterns: application of knowledge management theory to junior doctors training. Healthcare Computing 2004.
26. Coiera E: Communication systems in healthcare. Clin Biochem Rev 2006;27:89-98.
27. Orlikowski WJ, Gash DC: Technological frames: making sense of information technology in organizations. ACM Trans. Inf. Syst. 1994;12:174-207.
28. Berg M: The search for synergy: interrelating medical work and patient care information systems. Methods Inf Med 2003;42:337-344.
29. Bassil S, U.Reichert M, Bobrik R, Bauer T: Access Control for Monitoring System-Spanning Business Processes. InEnschede, Centre for Telematics and Information Technology, University of Twente, 2007:15.
30. Rinderle S, Weber B, Reichert M, Wild W: Integrating Process Learning and Process Evolution - A Semantics Based Approach .Book Series Lecture Notes in Computer Science 2005;Volume 3649/2005 252-267
31. Bernstein K, Bruun-Rasmussen M, Vingtoft S, Andersen SK, Nohr C: Modelling and implementing electronic health records in Denmark. Stud Health Technol Inform 2003;95:245-250.
32. Mulyar N, Pesic M, van der Aalst WMP, Peleg M: Declarative and Procedural Approaches for Modelling Clinical Guidelines: Addressing Flexibility Issues .Book Series Lecture Notes in Computer Science 2008 4928: 335-346.

Integrating Healthcare Ontologies: An Inconsistency Tolerant Approach and Case Study

Fahim Imam[1] and Wendy MacCaull[2]

Centre for Logic and Information,
Department of Mathematics, Statistics and Computer Science,
St. Francis Xavier University, Antigonish, Nova Scotia, Canada
{mimam, wmaccaul}@stfx.ca

Abstract. A major challenge for ontology integration is to effectively deal with inconsistencies that arise during the merging process. Because of the explosive nature of classical logic, the common strategy in existing merging tools is to choose between the contradictory pieces of information and maintain consistency. In many cases inconsistent information may be useful for intelligent reasoning activities. For example, in healthcare systems inconsistent information may be required to provide a full clinical perspective so any information loss is undesirable. In this paper we present a multi-valued logic based merging system that has inconsistency tolerant behavior and avoids information loss. As an application of the system in the healthcare domain, a result of merging a subset of two healthcare ontologies SNOMED CT® and ICNP® is presented.

Keywords: ontology integration, paraconsistency, multi-valued logic

1 Introduction

Originating in Philosophy, the term ontology was first adopted to computer science by the artificial intelligence community as "explicit specification of conceptualization" [1]. An ontology provides an effective way to represent a certain domain knowledge (e.g., regarding healthcare, molecular biology, business applications, etc.) by means of categorizing or organizing the concepts involved in the domain into a hierarchy and precisely specifying how the concepts are interrelated with each other in that domain.

The idea of ontology has become widely popular with a wide range of applications including software design, expert systems, database architectures, and, most recently, semantic web applications. Along with these generic applications, the importance of ontologies has been well established in the healthcare and bio-informatics community

[1] Supported by Natural Sciences and Engineering Research Council of Canada (NSERC-RCD) Graduate Fellowship

[2] Supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Award

in the past several years. Currently, the number of healthcare ontologies in various domains is growing very fast and an increasing number of them conform to various terminological standards. It has become a critical issue to share and reuse the combined but overlapping domain knowledge from the existing ontologies, especially those that conform to international terminological standards. One way to deal with this issue is known as ontology merging. This kind of integration is highly in demand where the goal is to generate a single coherent ontology that ensures the maximum reuse of knowledge from multiple source ontologies.

The significance of ontology merging or integration can be observed in the context of the Pan-Canadian Electronic Health Record (EHR) system. EHRs enhance the flow of information across multiple healthcare disciplines through the use of clinical terminologies and uniform language. Because of the diversity of various clinical terminologies, the challenge is to find a single terminology to represent all of the care providing disciplines. For instance, Canada has adopted SNOMED CT®[3] as the recommended clinical terminology for the EHR. However, the Canadian Nurses Association (CNA) recommends ICNP®[4] to represent nursing practice as Snomed CT® is more focused on the bio-medical perspective of healthcare. There are currently debates at the national level about whether Snomed CT® has enough representational capacity to effectively record nursing practices. As a result, CNA recommends that ICNP® and Snomed CT® collaborate to ensure that Snomed CT® is developed in such a way so as to effectively represent nursing practice in EHR. Merging or integrating ICNP® and Snomed CT® terminologies will be invaluable for the overall success of EHRs to provide extensive representational capacity. [2, 3]

A challenge for this kind of integration is to effectively deal with inconsistencies. Although the ontologies to be integrated are expected to be in a similar domain, they have their own distinctive perspectives specific to their own applications. The ensuing differences can lead to various kinds of inconsistencies during the ontology integration process. Generally, inconsistencies are considered to be unacceptable. According to [4] "all seem to agree that data of the form p and ~p for any proposition p cannot exist together and that the conflict must be resolved some how". This idea is, however, a consequence of the 'explosive' nature of classical logic. In classical logic, because of the 'principle of explosion', 'anything' (and therefore nothing useful at all) can follow, i.e., be inferred, from a set of inconsistent premises [5, 10]. As classical logic has been the main basis for computer science theory and practice so far, the existing ontology integration systems [6, 7] are based on maintaining consistency. However, enforcing consistency gives rise to valuable information loss by not allowing any inconsistent information in the system. Although enforcing consistency may have advantages in some cases, there are many cases where they could be useful to promote various intelligent reasoning activities such as those identified in [4]: natural processes of argumentation, information seeking, multi-agent interaction, knowledge acquisition and refinement, adaptation and learning and so on. Most importantly, inconsistent information is useful to evaluate the advantages and disadvantages of different perspectives or expert knowledge. In health care systems,

[3] Systematized Nomenclature of Medicine Clinical Term. http://www.snomed.org
[4] International Classification of Nursing Practice. http://www.icn.ch/icnp.htm

inconsistent information may be required to provide a full clinical perspective where no information loss is desirable.

In this paper we present a merging system along with a decision support mechanism that has inconsistency tolerant behavior and avoids information loss. In order to develop such a system we took a non-classical logic-based and multi-valued approach to retain the maximum possible information. The rest of the paper is organized as follows. Section 2 briefly introduces the common inconsistencies that can be observed in ontology integration and outlines an approach to deal with inconsistencies. The design and implementation of the new system are introduced and briefly discussed in Section 3. Section 4 presents a result of a sample merging of a subset of Snomed CT® and ICNP® ontologies as a case study in healthcare provision. We conclude this paper with directions for future work in section 5.

2 Ontology Integration and Dealing with Inconsistencies

Inconsistencies are inevitable for any kind of ontology integration. Usually, the ontologies are built without other ontologies in mind and different ontologies are developed by different people with different levels of domain requirements and expertise. Therefore, the most common kinds of inconsistencies occur because different ontologies may have the same terms with different meanings, different terms for the same concept and, different definitions for the same concept. The following example, motivated by Schlobach and Cornet [9], is provided to observe a typical scenario that causes inconsistencies in the case of ontology merging.

Let us consider the statements regarding 'Brain' in some ontology as follows:

1. $Brain \sqsubseteq CentralNervousSystem$	(a brain is a central nervous system)
2. $Brain \sqsubseteq BodyPart$	(a brain is a body part)
3. $CentralNervousSystem \sqsubseteq NervousSystem$	(a central nervous system is a nervous system)

A second ontology may conform to the 1st and 3rd statements of the first ontology but categorizes the concepts 'BodyPart' and 'NervousSystem' as disjoint as follows:

1. $Brain \sqsubseteq CentralNervousSystem$	(a brain is a central nervous system)
2. $CentralNervousSystem \sqsubseteq NervousSystem$	(a central nervous system is a nervous system)
3. $BodyPart \sqsubseteq \neg NervousSystem$	(a body part is not a nervous system)

Merging the two ontologies above would imply the contradiction that 'Brain is a body part and not a body part'. This kind of inconsistency is known as ontological inconsistency, where a concept is asserted as subclass of multiple disjoint concepts.

As mentioned earlier, there are many cases where information loss due to inconsistencies is not desirable. As classical logic fails to deal with inconsistencies, various non-classical logics have been developed to overcome its shortcomings. Formulas that are equivalent classically may be non-equivalent non-classically. This

provides an enriched set of logical operators such as variants of classical implication and negation operators. The paraconsistent logics [4, 5] are non-classical logics which we believe provide a suitable setting for our purpose. With paraconsistent systems, inconsistent information can be retained but the derivation of non-trivial inferences is possible. The contradictory propositions can co-exist safely without interfering with each other. The main motivations and the basic approaches for paraconsistent logics can be found in [10].

3 Design and Implementation

Merge two ontologies means building a new ontology from two existing ontologies with known concepts/classes and relations. The ultimate task is to organize and reuse the existing concepts and relationships to develop a single terminology which represents a maximized perspective of the domain knowledge. In this section we introduce the implementation issues, design principles, and the merging process involved in our ontology integration system. Refer to [17] for a detailed discussion.

Representation Language. To represent the ontologies, several choices exist among standard ontology languages [6] such as F-logic, KIF, OCML, LOOM, CycL, etc. The most recent development in ontology language is OWL [12], which is a standard ontology language defined by W3C. We consider OWL-DL [11] to be our representational language. Largely influenced by Description Logics (DLs), OWL-DL is designed to be expressively powerful and provides an efficient reasoning mechanism. This choice of using OWL-DL is a consequence of current concerns on semantic web applications, suitability for bio-medical ontologies, availability of tools support and standard Application Program Interfaces. Before merging, both of the source ontologies should be presented in OWL-DL language in our case. Another important reason for choosing OWL-DL is the availability of standard DL reasoners [11] which can facilitate automatic subsumption/classification of classes and consistency checking, thus providing implicit consequences of an ontology from the explicitly represented knowledge. For our merging system we used a standard classical DL reasoner called Racer Pro [13].

Ontology Specific APIs. An ontology-specific API often can save considerable programming effort for developing any ontology-based application. One such API is the Protégé-OWL API [14] which is an open-source Java library for OWL ontologies. The API comes with the Protégé Ontology editor package [15] developed by Stanford Medical Informatics group. The API provides various useful Java classes, interfaces, methods and other resources specific to OWL ontologies. It also provides optimized GUI interfaces and has the ability to connect with standard DL inference engines to perform reasoning based on standard DLs [14].

A Multi-Valued Logic and a Paraconsistent Reasoner. One of the easiest, yet effective ways to achieve paraconsistancy is to use multi-valued logic semantics. We are using a 4-valued logic called ALC4 [16] which is an extended version of standard description logic ALC, and suitable for OWL-DL based ontologies. The four truth values T, F, B and N in ALC4 are interpreted for four kinds of situations for a relation

between a pair of concepts (e.g., Bird – (isA) – FlyingAnimal) as follows: T (True) represents the situation where we know that the relation is true, F (False) represents the situation where we know that the relation is false, N (Neither True Nor False) represents the situation where we have no knowledge whether or not the relation holds, and finally, B (Both True and False) represents the situation we have contradictory information and the relation both holds and does not hold. In addition, the ALC4 logic [16] comes equipped with three kinds of implication operators namely material implication, internal implication and strong implication. *Material implication* is defined by means of classical negation and disjunction i.e., $a \mapsto b$ is definable as $\neg a \vee b$. Material implication in ALC4 is a limited version of classical implication as it does not satisfy modus ponens or the deduction theorem. The *Internal implication* cannot be defined by means of connectives; it does satisfy modus ponens and the deduction theorems but does not satisfy contraposition (i.e., if $A \rightarrow B$ then $\neg B \rightarrow \neg A$). *Strong implication* is stronger than the internal implication as it satisfies contraposition. The logic ALC4 allows simultaneous use of its three implications. Corresponding to these three implication operators, ALC4 allows three kinds of class inclusions in the context of OWL ontologies such as material inclusion, internal inclusion and strong inclusion. There is a paraconsistent reasoner called ParOWL for ALC4 logic. Refer to [16] for details about this reasoner and the semantics of ALC4 logic. As we want our system to be inconsistency tolerant, we pass the relevant set of inconsistent classes and their hierarchical information to the ParOWL reasoner which produces possible solutions from inconsistent situations.

Basic Design Principles. We believe that the following six issues are of fundamental concern for the development of an ontology integration system:

1. The merged ontology must preserve the different hierarchical views of the concepts found in the original sources – no matter how different/contradictory they might be (i.e., it should be tolerant to inconsistencies).
2. The merging system should be associated with an automated decision support mechanism which produces suggestions to permit the reuse of existing concepts from the sources and appropriately organizes the hierarchy for the resulting ontology.
3. The merging system should communicate with standard classical DL reasoners. This will facilitate automated computation for subsumption (or classifying taxonomy) and consistency checking from the explicitly specified concepts.
4. A paraconsistent reasoner should be associated with the merging system to deal with the set of classes that lead to contradiction in order to keep a meaningful hierarchy without losing the source knowledge.
5. The overall merging process should be semi-automated: human intervention by domain experts is required to commit (or discard) any automatically generated suggested tasks.
6. The merging system should provide enough GUI support that enables the domain experts to reason and perform a suggested task based on the visually accessible source and evolving ontology knowledge.

The Merging Process. Briefly, the overall process of our merging system involves the following main steps. The schematic view of the steps is presented in Fig. 1.

- ***Step 1.*** Identify the identical/equivalent concepts in the source ontologies with the help of domain experts.
- ***Step 2.*** Arrange the classes in the merged ontology based on the knowledge from the source ontologies.
- ***Step 3.*** Invoke a classical reasoner to produce an automated inferred hierarchy and to detect the ontological inconsistencies.
- ***Step 4.*** Invoke a paraconsistent reasoner for inconsistent situations.

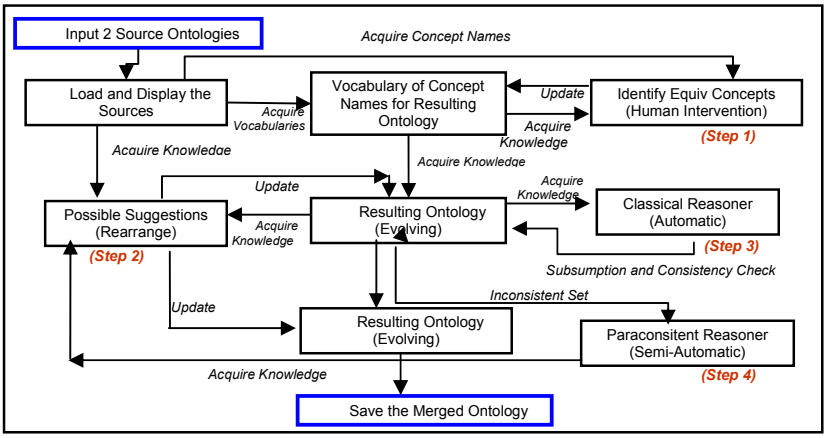

Fig. 1. The schematic view of the overall merging process.

It should be noted that after the step 1, the other steps do not proceed in a strictly linear fashion; rather, the activation of a particular step would depend on the evolving ontology. Schematically, we may think of the process more like Step 1→ Step 2 ←→ Step 3 → Step 4 → Step 2 etc. Assuming that both of the source ontologies are represented in the same language (currently our system only supports OWL-DL) the system first loads the source ontologies and displays them in a visually accessible way. The system guides the domain experts by providing automated suggestions throughout the process and focuses their attention on the likely points for action steps in Fig. 1. Refer to [17] for details about the steps and the overall merging process.

R(Ci, Cj)	O1	O2	M
disjoint(Ci, Cj)	F	T	B
overlap(Ci, Cj)	F	F	F
subclass(Ci, Cj)	T	F	B
superclass(Ci, Cj)	F	F	F
Sibling(Ci, Cj)	F	T	B

Fig.2. Relations Table to capture relationships between two classes Ci and Cj

The Relationships between Classes. The six basic kinds of relations between two classes or concepts that can be observed in an ontology are: equivalent, disjoint,

overlap, subclass, superclass and sibling. The relational information between two concepts is captured automatically in our system. The system then accumulates this information in a class-relations table such as the one in Fig.2, where *R*(*Ci, Cj*) returns the truth value for a relationship between two classes Ci and Cj from source ontologies O1 and O2; M represents the combined truth value for any relation returned by *R*(*Ci, Cj*) for O1 and O2.

We adopted four-valued logic semantics where the four truth values provide explicit representation for all possible situations that could be captured for a certain relation between a pair of concepts from the sources. Once the relational information is captured, the truth values of *R(Ci, Cj)* for O1 and O2 are combined according to the following rules:

- *If R(Ci, Cj) returns F for both O1 and O2 Then, M := F;*
- *If R(Ci, Cj) returns T for both O1 and O2 Then, M := T;*
- *If R(Ci, Cj) returns T for O1 and N for O2 (or vise-versa) Then, M := T;*
- *If R(Ci, Cj) returns F for O1 and N for O2 Then, M := F;*
- *If R(Ci, Cj) returns T for O1 and F for O2 (or vise-versa) Then M :=B;*

This choice of the value assigned to M reflects our fundamental principle, which is to preserve the maximum information. The truth value of M can assume one of 4 values so in the resulting ontology O3 there is a corresponding degree of belief about a relation.

Arranging the hierarchy. In the 2nd step (see Fig.1), our system looks at each pair of classes in the sources and determines the relations between them. For the pairs which are not equivalent/identical (i.,e., the relations between Ci and Cj in O3 when the equiv(Ci, Cj) relation returns F), our system automatically suggests the necessary changes to arrange the class hierarchy for O3 based on the generated relations table. The possible suggestions are as follows: (a) Add/remove Ci as a super/subclass of Cj, (b) Add/remove Cj as a sibling of Cj and (c) Add/remove the disjointness of Ci and Cj. Following the response from the user, the resulting ontology gets updated. This step is an iterative one, as each action from the user will be followed by a next suggestion, depending on the current hierarchy and progress of the evolving new ontology, until it terminates. Refer to [17] for further details on automated suggestion.

Invoking a Paraconsistent Reasoner. Our system identifies an inconsistency by determining if one ontology has F and the other has T (or vise-versa) for the returning truth values of R(Ci, Cj); the truth value B will be assigned to M (identifies a contradiction) for that relation. Let us consider merging the two Brain ontologies as discussed in Section 2. Consider the relation between the classes Ci = Brain and Cj = BodyPart. The relation table for these two classes can be captured as in the right hand table of Fig.2 which identifies contradictions in three of the relations, namely, the disjoint, subclass and sibling relations. The situation leads to the contradiction that 'Brain' is both a 'BodyPart' and not a 'BodyPart'. For cases such as this, the relevant set of classes with those relations that lead to inconsistencies from the previous step would be passed to the paraconsistent reasoner ParOWL. Based on the guidelines for ALC4 inclusions as discussed in [16], we assign each relation to one of three different OWL-DL files each consisting one of the three distinct inclusions. To invoke ParOWL, we pass the three files containing the ALC4 inclusions together with an

instantiation in the fourth file as in Fig.3. A different opinion from the domain experts about different inclusions can be used depending on their own view. Before invoking the reasoner, in the 4th step of the merging process, the new system provides an interface to capture and classify the type of class inclusions and produces four OWL files for ParOWL.

Material Inclusion (File 1)	**Internal Inclusion (File 2)**
isA(Brain, Thing) isA(BodyPart, Thing) ~isA(Brain, BodyPart)	isA(CentralNervousSystem, NervousSystem) isA(NervousSystem, BodyPart)
Strong Inclusion (File 3)	**Instantiation (File 4)**
isA(Brain, CentralNervousSystem)	isA(Brain, Thing) isAnInstanceOf(brain_1, Brain)

Fig.3. Inclusion Axioms for a Brain Ontology to invoke ParOWL

The reasoner generates a suggested hierarchy based on the sources with possible meaningful information. Based on the resulting hierarchy as presented in Fig.4 and considering that the ontology was classically inconsistent, the best possible solution for us is to combine P3 and P6, and conclude that Brain is subsumable by Bodypart, so we retain the inconsistent information and use multiple inheritances.

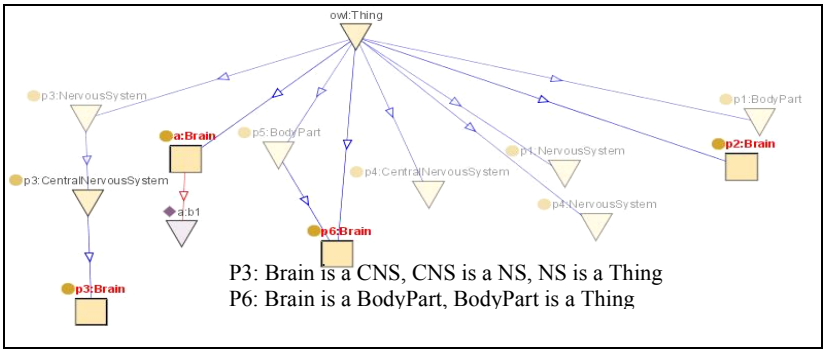

Fig.4. Resulting Brain Ontology produced by ParOWL reasoner

The basic design principles and overall process of our merging system is significantly different from existing systems. First of all, we support the argument that in many cases inconsistent information is useful and should be retained. Therefore, a practical reasoning mechanism for ontology merging must be inconsistency tolerant i.e., it can handle inconsistencies in a meaningful way and keep all possible information. Our system takes a paraconsistent logic approach using a four-valued logic to achieve inconsistency tolerance. ParOWL, a paraconsistent reasoner is used for this purpose. Second, our system takes an essentially logic-based and multi-valued approach in the overall merging process. Multi-valued logic is not only applied to capture the inconsistent set of classes and to reason about them but it is also used to observe and capture the structural information, identifying missing knowledge (when comparing

one source to another) as well as indicating the strength of suggestions. Taking the advantage of a four-valued logic to explicitly represent both unknown information (truth value N) or over defined, contradictory information (truth value B), provides the basis of producing sensible suggestions that are simpler to trace and reason.

4 Result of Merging ICNP® and Snomed CT®

In this section, the result of a sample mapping and merging of a small subset of ICNP® and SNOMED CT® ontologies are presented using our merging system. The concept 'Pain' was selected for merging, which is common in both source ontologies.

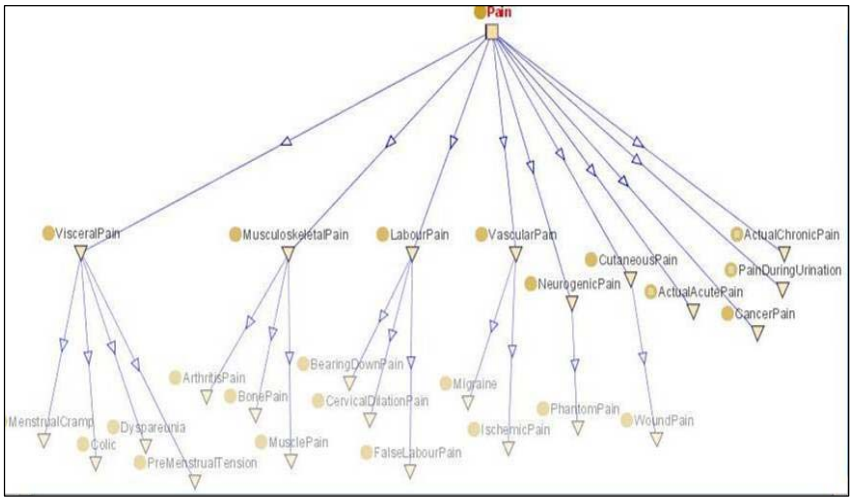

Fig.5. The subclass hierarchy of ICNP® 'Pain' concept. There are total 25 concepts involved in the hierarchy as displayed in the figure.

The Selected Subset. Fig.5 presents the ICNP® subclasses of the 'Pain' concept. Refer to this hierarchy to view the elements and specific paths when comparing the enhancements of the ontology after merging with corresponding SNOMED CT® classes in the next subsections. The hierarchy of the corresponding 'Pain_' class for SNOMED CT® is not presented here as it is considerably more complex and comprehensive to represent and visualize in a comparable fashion. However, the idea is to start merging the ICNP® concepts under 'Pain', one level at a time. So, the first level to be merged includes the concepts VisceralPain, MusculoskeletalPain, AcuteChronicPain, etc., which have the immediate root concept 'Pain' with the corresponding identical/similar concepts in SNOMED CT®.

The Result. This section presents the result of merging in terms of the overall enhancements in ICNP® and SNOMED CT® in the merged ontology. Table 1

presents the result of merging the subset of the ICNP® 'Pain' concept with SNOMED CT® in terms of the enhancements of the intermediate classes between 'Pain'and its sub classes. The first column contains the subclasses of the 'Pain' concept in ICNP® that were enhanced. The second column contains the added intermediate classes between the concepts 'Pain' and its particular subclass in the first column after merging with SNOMED CT®. If the intermediate class occurs in two or more rows, then the corresponding concepts in the first column are sibling classes and the intermediate class is their parent. The third column in the table represent the enhancements due to the further specializations of a particular ICNP® concept in the first column after merging with SNOMED CT®. If a rows in the second column of Table 1 contains two or more comma separated class names, the intermediate classes are related with superclass relations consecutively from the left to the right (from a more general concept to a more specific one). Note that, the original sources of the intermediate classes are obviously from SNOMED CT®. The comma separated class names in the third column are subclasses of the corresponding concept in the same row of the first column.

Table 1. The enhanced classes of ICNP® 'Pain' concepts from SNOMED CT after merging.

ICNP® Concept	Added Intermediate Classes from Snomed CT®	Added Subclasses from Snomed CT®
VisceralPain	Pain_Finding_At_Anatomical_Site	N/A
Musculoskeletal Pain	Pain_Finding_At_Anatomical_Site	N/A
LabourPain	Specific_Body_Function_ Causing_Pain, Obstetric_Pain	N/A
NeurogenicPain	Pain_Finding_At_Anatomical_Site	N/A
ActualAcutePain	Finding_Of_Pain_ Pattern	N/A
ActualChronicPain	Finding_Of_Pain_Pattern	N/A
Colic	Pain_By_Sensation _Quality	N/A
Dyspareunia	N/A	Deep_Pain_On_Intercourse, Painful_Orgasm, Superficial_Pain_On_ Intercourse
BonePain	N/A	Axis_Pressure_Pain, Bony_Pelvic_Pain, Bone_Tenderness, Clavic_Pain (and 12 more)
FalseLabourPain	Specific_Body_Function_ Causing_Pain, Obstetric_Pain	N/A
Migraine	Alimantary_Tract_Pain_Due_To_ Vascular_Insufficiency	N/A

Based on Table 1, the ICNP® 'Pain' concepts in the first column have been enhanced by a total 6 distinct intermediate classes from SNOMED CT® as listed in the 2nd column. Also, two of the classes in the 1st column have been enhanced by a total of 7 subclasses added from SNOMED CT® as listed in the 3rd column. The SNOMED CT® has been enhanced in a similar fashion with total 9 classes from ICNP® (See [17] for details). However, there were no new intermediate classes discovered after merging with ICNP® for SNOMED CT®.

So far, the results represent enhancements in the merged ontology in terms of added sub classes (in both ICNP® and SNOMED CT®) or intermediate classes (in

ICNP®). The truth value N was used to represent the situations where some class was missing in ICNP® compared to Snomed CT® or vise-versa. Table 2 summarizes the enhancements in the merged ontology due to the different hierarchical perspectives of certain classes in ICNP® when they are merged with corresponding SNOMED CT® concepts. Based on Table 2, if we count the distinct number of classes listed in the 2nd column, ICNP® has been enhanced by a total of 11 new classes.

Table 2. Enhanced classes in ICNP® due to different perspectives in Snomed CT®

ICNP® Concept	Added Super-Classes from SNOMED CT
VascularPain	Clinical_Finding, Disease, Disorder_Characterized_By_Pain
MenstrualCramp	Clinical_Finding, Disease, Disorder_Characterized_By_Pain
Dyspareunia	Clinical_Finding, Clinical_History_And_Observation_Findings, Functional_Finding, Finding_Related_To_Sexuality_And_Sexual_Activity, Sexuality_Related_Problem
Migraine	Clinical_Finding, Disease, Disorder, Disorder_Characterized _By_Pain, Headache_Disorder, Vascular_Headache
IschemicPain	Clinical_Finding, Disease, Disorder_Characterized _By_Pain
PhantomPain	Clinical_Finding, Disease, Disorder_By_Body_Site, Disorder_Of_Body_System, Disorder_Of_The_Peripheral_Nervous_System

For the case above, note that SNOMED CT® supports multiple inheritance where the classical OWL-DL ontologies (e.g. ICNP®) are based on simple inheritance. In this situation, the truth value B was used to specify that a certain hierarchical relation was both true (T) and false (F), as a same class cannot have two superclasses in traditional OWL-DL. The truth values T and F were first combined into B and the 4-valued reasoning was employed to keep both of the hierarchy to preserve the different perspectives. Refer to [17] for further details.

Finally, the result of overall merging shows that 32 new concepts have been added to the original 25 'Pain' concepts of Figure 1; this indicates that the process of integration leads to a significant enhancement as compared to either source.

5 Conclusion and Future Work

We took an inconsistency tolerant approach of merging Ontologies so that different perspectives can be retained without loosing any meaningful information. The result of merging the 'Pain' subset of ICNP® and SNOMED CT® indicates that the merged ontology became much more granular and detailed than any of the sources. Different perspectives were retained effectively using the multi-valued paraconsistent approach.

A more comprehensive understanding of suitable non-classical implications and an automated procedure to assign facts about various inclusions to the appropriate input files of ParOWL and reason with them is currently under investigation. Future work includes making the system more flexible, efficient and applicable for large scale Ontology development. In particular, developing a more advanced and generic paraconsistent reasoner, employing the parallel high-performance computing techniques for merging large ontologies and necessary requirements for realistic tools which permit reasoning with multiple views or incomplete information are directions for future work. It is hoped that this research will form the basis for the strong

recognition of inconsistency tolerance and paraconsistent logics, not only for existing ontology merging tools, but also for developing any semantic web and other ontology based applications.

References

1. Gruber, T.: An ontology for engineering mathematics, Proc. of the 4th International Conference on Principles of Knowledge Representation and Reasoning, pp. 258-269, Germany (1994)
2. Imam, F.,T., MacCaull, W., Kennedy, M.,A.: Merging Healthcare Ontologies: Inconsistency Tolerance and Implementation Issues, Proc. of 20th IEEE International Symposium on Computer-Based Medical Systems (CBMS'07), pp. 530-535, Publisher: IEEE Computer Society Washington, DC, USA (2007)
3. William, G.: Cross-Mapping Between Three Terminologies With the International Standard Nursing Reference Terminology Model, International Journal of Nursing Terminologies and Classifications (2006)
4. Bertossi, L., Hunter, A., Schaub, T.: Introduction to Inconsistency Tolerance, Lecture Notes in Computer Science., Vol. 3300, Springer Berlin/Heidelberg, pp. 1-4 (2005)
5. Hunter, A.: Paraconsistent Logics, Handbook of Defeasible Reasoning and Uncertainty Management Systems, Vol. 2, Kluwer Academic (1998)
6. Su, X., Lars, I.: A Comparative Study of Ontology Languages and Tools, Lecture Notes in Computer Science, Vol. 2348, Springer Berlin / Heidelberg (2002)
7. De Bruijn, J., Martin-Recuerda, F., Manov, D., Ehrig, M.: State-of-the-art survey on Ontology Merging and Aligning, a Report of SEKT: Semantically Enabled Knowledge Technologies (2004)
8. Huang, Z., van Harmelen F., Teije, A.,T., Groot, P., and Visser, C.: Reasoning with Inconsistent Ontologies: A General Framework, EU-IST Integrated Project (IP) IST-2003-506826 SEKT (2005)
9. Schlobach, S., Cornet, R.: Non-standard reasoning services for the debugging of description logic terminologies, Proceedings of the 18th IJCAI, Mexico (2003)
10. Priest, G., Koji, T.: Paraconsistent Logic, The Stanford Encyclopedia of Philosophy, E., N., Zalta (ed.) (2004) http://plato.stanford.edu/archives/win2004/entries/logic-Paraconsistent
11. Horrocks, I., Patel-Schneider, P.,F., McGuinness, D., L., Welty., C.,A.: OWL: A Description Logic Based Ontology Language, Logic Programming Book, Lecture Notes in Computer Science, vol. 3668, Springer Berlin / Heidelberg (2005)
12. Smith, M., K., Welty, C., McGuinness, D., L.: OWL web ontology language guide, W3C Recommendation (2004)
13. Racer: A Semantic Middleware for Industrial Projects Based on RDF/OWL, A W3C Standard. http://www.sts.tu-harburg.de /~r.f.moeller/racer
14. Protégé-OWL API. http://protege.stanford.edu/plugins/owl/api
15. The Protégé Ontology Editor and Knowledge Acquisition System. http://protege.stanford.edu/
16. Ma, Y., Lin, Z., Hitzler, P.: "Paraconsistent Reasoning with OWL - Algorithms and the ParOWL Reasoner", Technical Report, AIFB, University of Karlsruhe, Germany (2006)
17. Imam, F., T.: An Inconsistancy Tolerant Approach to Ontology Merging, M.Sc Thesis, St. Francis Xavier University, Library and Archives Canada, Canada (2008)

Integrating Humans, Devices, and Events in Clinical Workflow Processes

Jan-Christian Kuhr[1], Jan Pretzel[1],
Dierk A. Vagts[2], and Lachlan Aldred[3]

[1] GECKO mbH, 18057 Rostock, Germany,
{jku,jpr}@gecko.de
[2] Universität Rostock, Klinik für Anästhesiologie, 18057 Rostock, Germany
dierk.vagts@med.uni-rostock.de
[3] Queensland University of Technology, Brisbane Qld 4000, Australia
l.aldred@qut.edu.au

Abstract. Clinical workflow processes do not only strongly rely on human activities but may also depend on events that trigger the execution of work items. We describe a novel approach that integrates human tasks, devices, and events into a single process model. As a sample workflow, we discuss the patient's admission to an intensive care unit. Given an infrastructure of loosely-coupled disparate systems, such as sensor devices and clinical information systems, process integration is accomplished by adopting as well as extending an open-source messaging middleware that is known as JCoupling. We argue that JCoupling in conjunction with sensor technology allows for the replacement of manual, computer-related tasks by semi-automatic message-driven interactions. Our approach is based on the integration of JCoupling with the forthcoming release of the workflow management system YAWL. This ensures explicit support for resource-centric workflow patterns and thus lends additional power to our proposal.

Key words: Process modelling; workflow management; context-aware healthcare processes; integrating healthcare processes with electronic medical records.

1 Motivation

Modern hospitals and clinics are extremely busy working environments, creating hundreds, or perhaps thousands of cases every day. Patients, technical devices, physicians, nurses, and other hospital staff constantly create events. Some of these events are urgent while others don't mean anything unless they coincide with another event, in which case they may mean quite a lot. This places high demands on clinical organisation as well as processes. With rising costs, and increasing expectations for speed and quality of service by our communities, hospitals and clinics are under pressure to make their processes more efficient and timely. Here workflow systems can play an important role.

Our main research objective is to integrate workflow management systems (WfMS) with Auto-ID systems such as, e.g., RFID in order to support clinical processes by replacing human-computer interactions with sensor-computer

interactions. Furthermore, we investigate the applicability of the messaging middleware JCoupling, which is currently in a proof-of-concept status, for real-life business scenarios. We propose a system architecture that is intended to be implemented as part of a pilot project that is currently being carried out in close collaboration with the Clinic for Anaesthesiology and Intensive Care at Rostock University Hospital. The university operates a central SAP i.s.h.med hospital information system (HIS). Earlier this year, a COPRA patient data management system (PDMS) was introduced to the hospitals intensive care unit (ICU). HIS and PDMS play important roles in the clinic's everyday business routines. In the following, we extend the usual integration of human and non-human resources by explicitly taking into account events as well as sensor-based devices that generate those events.

2 Case Study

As a suitable case study (and reference workflow) we examined the patient admission at ICU, which is an everyday occurrence. Currently this process requires *manual* interaction with the hospital's HIS before other work items may take place. However, due to a systemic lack of timely resources, in practice the admission process is often unnecessarily slowed down. Hence, this workflow step has been identified as a potential candidate for improvement. The current proposal addresses this challenge by introducing a message-based integration of workflows and events.

The process is represented as a Coloured Petri Net (CPN) which has been created using CPN Tools[1] (Fig. 1). CPNs are well-suited for modelling of healthcare processes [6], [7]. They also integrate well with the workflow language YAWL (see sec. 3). The model clearly demonstrates the close interdependence of manual tasks (e.g. `Transport`, `Connect Devices`), human-computer interactions (e.g. `Synchronize WfMS and HIS`), device-related work items (e.g. `Move to Final Position`) and message-based tasks (e.g. `Register With PDMS`). Note that multiple roles of staff are involved (such as nurses, doctors, anaesthesists). Furthermore, work items that do not necessarily depend on one another may be executed concurrently. The admission process usually commences when a patient is being released from the operating theatre and fetched by ICU staff.

We give type definitions that are utilized by many places together with their significance: `colset SENSOR = int` (number of an auto-ID device), `colset MESSAGE = string` (message that is produced by a sensor), `colset Role = string` (a resource's role, e.g., doctor, nurse), `colset Roles = list Role` (a resource may have more than one role), `colset RES = product Role * Name` (human resource), `colset PAT = string` (patient). Among the variables that are used in many arc expressions are these: `var p : PAT`, `var b : BED`, `var s: SENSOR`.

2.1 Characteristic Process Features

In this section we comment on some characteristic process features.

[1] http://wiki.daimi.au.dk/cpntools/cpntools.wiki

`Synchronize WfMS with HIS.` In order to enable semi-automatic relocation with the SAP system at a later process step, this transition synchronizes the workflow with i.s.h.med to keep a copy of relevant patient data in the WfMS (such as *patient ID*, *old location ID*).

`Relocate Inside HIS.` Prior to connecting a patient to the PDMS (which can only take place at the final patient position) it is necessary to perform a relocation request within SAP which will update the patient record with the new ward's location ID. To this end, a sensor device such as an RFID reader captures the entering of the patient bed right at the entrance of the ward, which in turn initiates a relocation request with the SAP system. The complex sensor-to-workflow and workflow-to-HIS interactions that are involved in this transition are described in Section 4.

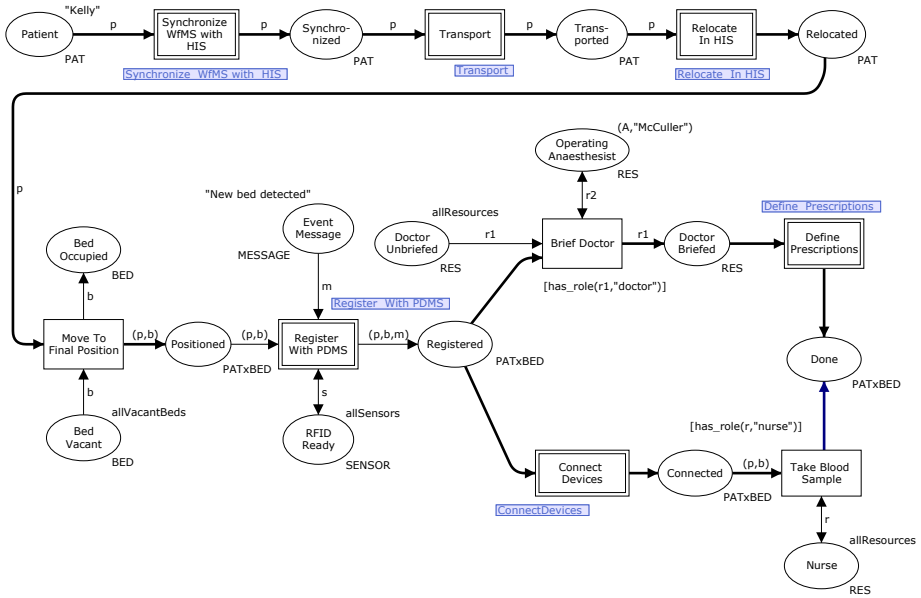

Fig. 1. Simplified CPN model of a patient's admission in an ICU setting.

`Register With PDMS.` To be enabled, this substitution transition requires three tokens at the input places, representing a patient in a bed, a ready-to-detect RFID reader and a message to be sent. The associated subnet is responsible for performing all tasks that couple the event to the WfMS, transform the event into a command message, and send this message to the PDMS system. Eventually, PDMS performs a message-driven patient registration request and comes back with an acknowledgement response.

`Brief Doctor.` This transition represents a task that is purely human in nature. The guard expression states the required role of the resource that is to be briefed being a *doctor*. In practice one might wish to place further conditions on the

resource, such as personal skills or organisational requirements. The presence of human-centric work items in a workflow calls for support of the so-called resource perspective, a subset of the well-known workflow patterns [3][4].

3 Selected Technologies

This proposal aims to address the challenges of the workflow introduced in Section 2. Our proposal builds on a selected set of technologies and best-practices. These include:

1. **Sensor Devices.** We envision a scenario where location-aware sensor devices, such as RFID readers or barcode scanners, are placed at ICU key positions. For the sake of simplicity, we hereafter assume an RFID scenario. However, our approach is also applicable for other comparable sensor technologies. All patient beds as well as human resources carry RFID transponder tags that allow for easy and uncomplicated identification. At the ward's entrance an RFID reader captures incoming patients by semi-automatically detecting their transponder tags. Likewise, RFID readers in close PDMS vicinity generate a signal whenever a bed is positioned at this location. Both signals serve to initiate message-based communication with clinical information systems and form an important part of the *admission* workflow.
2. **Sensor Backend System.** We intend to build the sensor backend system on Sun's RFID 3.0 Software architecture[2], which is particularly suitable for integrating loosely coupled sensors into a dynamic network. Furthermore, by providing a JMS-based messaging gateway, it integrates well with other components of the system.
3. **Messaging Middleware using JCoupling.** For process integration we generally use a messaging approach [8] that is based upon a novel communication platform, known as JCoupling. JCoupling is a language for expressing event sources, event sinks, and event filters within workflows. It encapsulates state of the art middleware technologies, exposing them at the workflow level. We intend to develop the reference implementation of JCoupling into a production-ready solution for event handling.
4. **Integration with YAWL.** YAWL is a state-of-the-art workflow language. It was inspired by CPNs but has a specific semantics more suitable than CPNs for the modelling and execution of complex business processes. YAWL is also a powerful open source WfMS. We have adopted YAWL as a workflow language of choice because it exhibits the following features that we consider important in our application domain: (1) expressiveness in terms of support for workflow patterns [2] (2) strong support specifically for resource workflow patterns (3) CPN-like semantics and integration with tools like CPN Tools (4) flexibility by means of so-called worklets (5) integration with JCoupling. To integrate YAWL with JCoupling, we define a type of message receipt workflow task such that: (i) upon enabling, the task registers a one-off message consumption filter with JCoupling, defined as part of a task decompo-

[2] https://sun-rfid.dev.java.net/SJS_RFID_Software.html

sition; (ii) the task then waits until the filter returns a match; (iii) should the task be cancelled before a match is found, the filter is withdrawn.

5. **Support for Human-Centric Workflow Patterns.** We plan to build the system on the forthcoming release of YAWL, hereafter referred to as YAWL 2.0, of which a beta version has become available by end of May 2008. YAWL 2.0 addresses human-centric processes by explicitly supporting many of the workflow patterns that are related to the so-called resource perspective.

4 Integration of Events, Devices and Information Systems

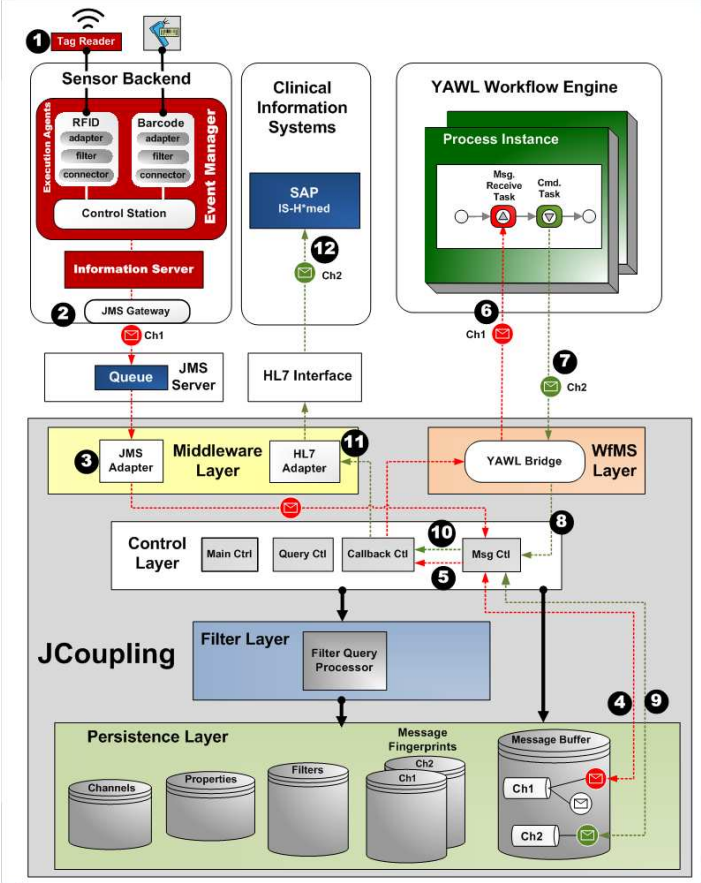

Fig. 2. End-to-end integration of events, WfMS, and clinical information systems, exemplified by the case of an RFID reader as primary message source and a SAP i.s.h.med HIS.

The business processes under consideration require an end-to-end integration of events on the one side and central as well as peripheral clinical computer systems such as HIS and PDMS, on the other. To this end we adopt an extended version of the open-source middleware architecture JCoupling [1]. JCoupling will

be specifically adapted to integrate with the new YAWL workflow engine. On the other hand, JCoupling has to be extended to allow for communication over HL7-based channels, i.e. to exchange HL7 messages with clinical information systems. A simplified message flow is shown in Figure 2.

Principle. In principal, an end-to-end integration of sensor network and clinical information systems comprises *two* communications, namely (a) propagation of an event message to the WfMS (Sensor Backend → YAWL) and (b) communication of a command message to the peripheral clinical system (YAWL → Clinical Information Systems). Whereas the first channel (`ch1`) is a JMS queue, the second is related to HL7, a common interoperability messaging standard in healthcare (`ch2`). We exemplify this process by assuming that the initial message source is an RFID reader and the final message sink is an HIS such as SAP i.s.h.med .

Message Flow. **(1)** An RFID reader detects a signal from a transponder and forwards this event to the Event Manager. **(2)** Event Manager converts this signal into an JMS message that is being forwarded to the message queue of a JMS server (OpenJMS). **(3)** JCoupling's JMS adapter periodically polls the JMS queue and consumes incoming messages off the queue. **(4)** The message is identified to be associated to (`ch1`) (which has been defined previously) and stored into JCoupling's persistence layer. **(5)** JCoupling's query processor finds that the fingerprint of the message matches a filter that a YAWL process instance has previously registered. The associated message is retrieved from the database and passed onto the callback controller (`callback ctl`). **(6)** `callback ctl` correlates the event message via the `YAWL Bridge` with a message receive task of the correct YAWL process instance. Upon receipt of the message, the receive task completes and workflow control is passed on to a command task. **(7)** The latter recognizes that an action has to be taken in response to the incoming message and thus enters into a request - response communication with the remote SAP system. Now the situation is reversed: YAWL acts as a message source and forwards a command message to JCoupling via (`ch2`). **(8)** JCoupling persistently stores the message in a message buffer, thus ensuring time decoupling of sender and receiver. **(9)** Upon availability of the SAP endpoint, `callback ctl` tries to forward the message to the HL7 interface of the Clinical Information System. **(10)** A specifically designed HL7 adapter, as part of JCoupling's middleware technology layer, transforms the command into a valid HL7 message that can be processed by the endpoint. **(11)** The SAP system performs the request by processing the incoming command message and sends an `ACK` message back to the WfMS upon successful completion. For the sake of simplicity, we do not show the flow of the response message in Figure 2.

5 Conclusions and Future Work

We have described a proposal how various aspects of modern hospital's business processes such as humans, devices, and events, may be integrated into a single workflow execution architecture. As a starting point, we have investigated the

procedure of patient admission at ICU for which we have created a simple CPN model. Future work will have to implement and test the applicability of our proposal on-site at the Rostock ICU, turning it into a real-life application and investigating the impact on workflow optimization. Further objectives may be to extend this approach to other clinical workflows as well as to investigate whether we can make workflows still more context-aware by adapting them to events that are generated by Auto-ID sensors. Also, it might be interesting to replace YAWL by other workflow languages such as BPEL, thereby opening our approach to those application domains where a BPEL-based solution may be required.

Acknowledgements. This work has been supported by the German Federal Ministry of Economics and Technology under grant KF0478101SS7. For fruitful discussions we are indebted to Markus Band, who has also carried out a detailed on-site analysis of the patient admission procedure which served as a basis for the CPN model of this proposal. Furthermore we would like to thank the head of the Department of Anaesthesiology and Intensive Care, Gabriele Nöldge-Schomburg, for making this research project possible at the Rostock University Hospital. We are especially grateful to Arthur H.M. ter Hofstede and the YAWL team in Brisbane for their collaboration towards the adaptation of YAWL to our project's purposes. Finally, we thank Michael Westergaard, Aarhus, for making available to us an internal release of CPN Tools 2.3 and Jonathan MacGill, Rostock, for correcting phrases and punctuation in the English text.

References

1. Aldred, L., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M.: Communication Abstractions for Distributed Business Processes. Proc.Int.Conf. on Advanced Information Systems Engineering, Trondheim, Norway (2007). For the projects source code see: http://sourceforge.net/projects/jcoupling/
2. Russell, N., ter Hofstede, A.H.M. , Edmond, D., van der Aalst, W.M.P.: newYAWL: Achieving Comprehensive Patterns Support in Workflow for the Control-Flow, Data and Resource Perspectives. BPM Center Report BPM-07-05, BPMcenter.org. (2007). Also see: http://www.yawlfoundation.org.
3. Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P., Edmond, D.: . Workflow Resource Patterns: Identification, Representation and Tool Support. Advanced Information Systems Engineering: 17th International Conference, Porto (2005)
4. van der Aalst, W.M., ter Hofstede, A.H., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14 (1), pp. 5-51 (2003).
5. Jørgensen, J.B., Bossen,C.: Executable Use Cases: Requirements for a Pervasive Health Care System. Software 21 (2), pp. 34-41 (2004)
6. Jørgensen, J.B., Lassen, K.B., van der Aalst, W.M.P.: From task descriptions via colored Petri nets towards an implementation of a new electronic patient record workflow system. Int. J. on Software Tools for Techn. Transfer 10 (1) (2008)
7. Mans, R.S., van der Aalst, W.M.P., Bakker, P.J.M., Moleman, A.J., Lassen, K.B., Jørgensen, J.B.: From Requirements via Colored Workflow Nets to an Implementation in Several Workflow Systems. In K. Jensen (Ed.), Proceedings of the Eighth Workshop on the Practical Use of Coloured Petri Nets and CPN Tools, DAIMI 584, pp. 187-206, Aarhus, Denmark (2007)
8. Hohpe, G., Woolf, B.: Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley (2004)

Session-Aware Clinical Information Systems

Øystein Nytrø[1,3] and Inger Dybdahl Sørby[1,3] Ole A. Alsos[1,3]

[1] Department of Computer and Information Science
[2] Faculty of Medicine
[3] The Norwegian EHR Research Centre
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
{nytroe, ingerdyb, oleanda}@idi.ntnu.no

Abstract. Modern health information systems must be able to represent and reason about work-processes, roles, tasks and interaction in order to provide relevant and timely support. However, commonly used models of work and interaction does not address the details of everyday clinical collaborative communication. If we are to design clinical information systems that support processes, we have to deal with the reality as perceived by the clinician: A puzzle of information and communication about patients, decisions and events. We have identified that the dialogue between one or more users and systems is important to capture and support.
This paper introduces the *session* as a representation of such dialogues. The sessionis a sharable, referable, persistent representation of the dialogue between persons and systems. Features of systems that support sessions are explained through example scenarios. The analysis proposes a set of requirements for session-aware clinical information systems.

Key words: decision-support, care process, computer-supported collaborative work, activity-maps, session

1 Introduction: Supporting the multitasking clinician

Work in a hospital ward involve many clinicians collaborating in many processes concurrently. The actors need long training, impressive mental capabilities and intimate mutual knowledge about objectives, needs and intentions in order to perform this intricate work. Introducing information systems in this fine-tuned dance is like putting elephants in the ballet. Today's information systems are hardly able, nor willing.

Context-aware, mobile, systems is part of a future solution. They depend on their observable surroundings:

Location and time along with rosters, meeting plans can trigger the system's profile or interface.

Role and preferences of the current user may enable individually adjusted or role-specific system behaviour.

Devices, tools and patients in the proximity may drive lookup, communication and reminders.

Events and conditions in information systems , for example the arrival of a test result out of normal range, can trigger reminders and task support.

However, the majority of relevant events for the clinician are not included in this physically focussed context, — the essential activities and events occur between people and in their heads. In order to uncover this part of the context, we have developed methods for analysis and observation of ward collaboration [1]. The resulting models serve as fundament for process-aware context representations and more process-enabled interfaces to clinical information systems.

Our objective is to make pervasive interfaces to health information systems that are aware of the user's communication- and collaboration needs. This leads to a requirement that the system must be able to represent and reason about the collaborative, communicative processes.

Many of these processes can be viewed as instantiations of care plans, whether they are implicit or explicit. The choreography of collaboration and patient treatment is a enactment of a *treatment plan* but driven by external events and interference from other, concurrent treatment plans in execution. In principle, these plans are made individual by combining and adjusting local or other recommended guidelines for treatment.

Our argument can be summarized by a sequence of statements:

- "The success of clinical decision support systems requires that they are seamlessly integrated into clinical workflow" [2].
- Clinical workflow support must be part of the clinical information system.
- Clinical information systems must be efficiently usable at the point of care.
- Usable mobile, point-of-care, clinical systems must be aware of, and support, the collaborative communication processes among systems, care providers, patients and other actors.

The collaborative communication processes are driven by common and mutual knowledge about plans, guidelines, objectives, expected outcomes, intentions, roles and responsibilities. Our hypothesis is that clinical decision support can only be realized in clinical systems that represent and use information about:

1. Explicit plans and activities that is the context of work.
2. Roles and responsibilities of actors.
3. Outcomes and objectives of actions and actors.
4. The objective of information (as communicated by someone).

From a user point of view, the "representation of interaction" is not important, but the resulting *behaviour* of the information system is. In order to be concrete, the next section presents a scenario involving a hypothetical mobile interface to a clinical information system. The scenario is illustrated by an activity-map which shows session handover, suspension, and reference.

Our objective in this paper, is to explain the importance of supporting the dialogue between users and system(s). Furthermore, we define the *session* as a

representation this interactive collaborative dialogue that can be supported by a context- and guideline-aware information system interface. Finally, we propose a set of requirements that a representation of a session should fulfill.

2 A detailed scenario

The following scenario, previously [3] developed in cooperation with physicians and also supported by empirical data from observations in hospital wards, will be used to explain some features of a session-aware CIS. The litera are added for reference only, they are not intended as rigid subdivisions of activities!

A *It is early afternoon in the Coronary Care Ward. Dr. Dorman is almost finished for the day and her last task is to write a discharge summary for patient Patterson who will be transferred to a nursing home later in the afternoon. She opens a template session with the suitable tools for making the discharge summary on her office computer.*

B *Nurse Nathan is preparing for the medication round in the wards separate medication room, on the local computer, and reviews patient Andrews prescriptions. He wonders whether the Warfarin dose (an anticoagulant) is correct, and sends a query, with reference to the relevant session, to the responsible doctor on duty (Dr. Dorman).*

C *Dr. Dorman suspends work on the discharge summary, and opens her view of the session with patient Andrews current status and medication plan referred in the query from the nurse. Meanwhile, nurse Nathan continues medication with other patients.*

D *After assessing the information, Dr. Dorman is about to change the medication dose, but then she is called by nurse Kim to patient Trent who has had a ventricular tachycardia. She has to go there immediately, leaving the medication of patient Andrews unfinished, i.e. suspending the session. On her way, she opens the referred session on patient Trents vital data in her PDA. That session was initiated by nurse Kim.*

E *As soon as Dr. Dorman finished treating patient Trent, and leaves him with Kim again. On her way back, she opens the session on Andrews medication on her PDA, changes the Warfarin dose, and notifies the responsible nurse (Nathan).*

F *Nurse Nathan finishes his current medication task, and attends patient Andrews, notes the change of medication and finishes medication of Andrews. Dr. Dorman resumes the session on patient Pattersons discharge summary, but instead of finishing the session, Dr. Dorman decides to hand it over to Resident Roberts, who is about to go on call for the afternoon.*

G *During a handover meeting, Dr. Dorman orally informs Dr. Roberts about the discharge summary to be written, and transfers responsibility on her PDA. Later, in his office, Dr. Roberts opens the discharge session and finishes it.*

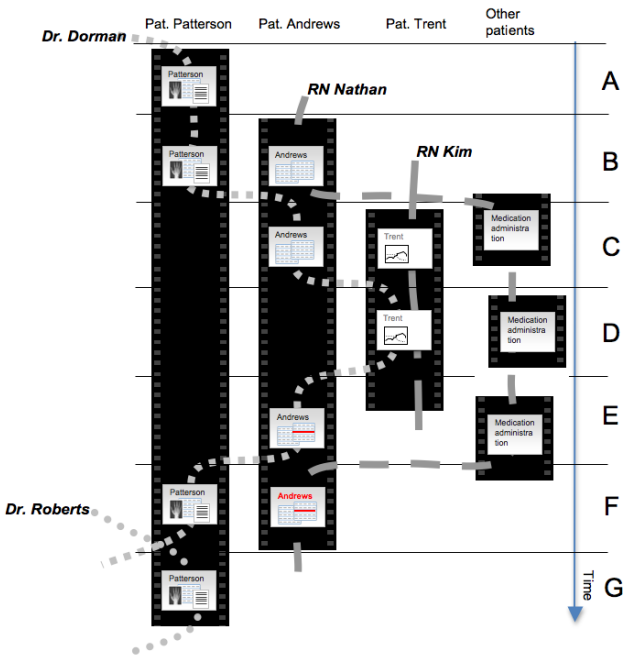

Fig. 1. Actors attending sessions in different contexts in the coronary ward

Figure 1 illustrates the various parts of the scenario in an activity map. The time runs downward. Each "film strip" represent a continuous session of work within a delimited subprocedure or task in a specific context. Here, the contexts are represented by patient names. A non-black frame (a computer screen snapshot) means that someone is actually looking at, - or working on the session. A black film represent a session that no-one is working with it, - it is suspended, open, but still referable. A session occurs in a context, a slowly changing set of information describing a problem, or as in this case, a patient. A context may have more than one session open at a time, but this is not shown here. An example of multiple, parallell sessions would be the nurse attending medication, while radiologists discuss the latest MRI findings. The attention of actors are shown as dashed lines, weaving between different sessions. We call these actor-trajectories. Actors may share one session, and they may have different views of the same session. Eg. RN Nathan has access to the medication plan of patient Andrews, but Dr. Dorman will additionaly see the prescription interface in the same session.

Reference to sessions and handovers are not shown explicitly on the figure due to visual overload, but explained in the scenario. A handover is a message (with intention "handover - take responsibility" and a reference to the specific session.) We usually draw messages as arrow-like bars, from a sending actors

actor-trajectory to a receiving actor, - annotated with the intention and content between. The handover between Dr. Dorman and Dr. Roberts would look as shown in figure 2. Dr. Roberts is not interrupted, buth the handower is waiting for his attention. If the message is interrupting, it would be shown as a horizontal arrow between two actor-trajectories, because it would be a synchronous event (ie. approximately at the same point on the time-axis for both actors.)

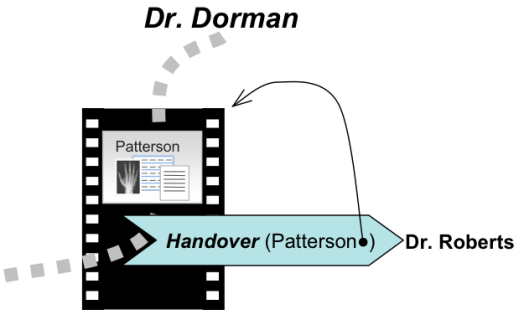

Fig. 2. A handover-message

3 Related Work

3.1 Networking and telecommunication

The session concept is inspired by session layer of the OSI model of computer communication[1], appearing as ISO standard 8326 (and available as ITU X.225 [4]). The session layer is responsible for a time-delimited dialogue between two specific systems, in which the dialogue is stateful in the sense that at least one of the partners preserve a history of the interaction so far. This means that a session can be set up, started, delayed, restarted, taken down etc. In general telecommunication terminology, a session is the channel between two users of the telecommunication system, for example a phone conversation. In web terminology, a session is a time-delimited, stateful, interaction between an actor and a web-server that appears to be something more than simple fetching of unrelated webpages. Session mobility and preservation come as natural consequences of the separating between session and the underlying layers: The session should be independent of underlying physical infrastructure.

[1] The OSI (Open System Interconnection reference model) subdivides the functionality/responsibility of communication into seven layers: Physical, data link, network, transport, session, presentation and application.

3.2 Pervasive computing, CSCW and context-awareness

Support of multiple sessions for single users were mainstream already with the X windows system, which separated application window from display and input devices, and display from process, machine and user. This made sharing and interaction with sessions straightforward. Many examples of computer-supported collaboration and pervasive frameworks relied on this functionality, one representative example is SharedX [5]. The literature on CSCW and ubiquitous computing that addresses shared, mobile work, is overwhelming. Relevant work includes session migration [6], pervasiveness [7] and shared workspaces [8]. Surprisingly much of this has been technology in search of applications, - of which there have been precious few. Only recently has some of this work been seen in the light of clinical work. Bardram and Bossen [9, 10, 11] lay much of the necessary groundwork, but does not fully address the challenges in the interface between decision support and activity-based collaboration.

Context-awareness of a user-oriented system is the capability to sense, represent and (re)act on the system's and user's environment. A context-unaware system must rely on the user to provide all the background knowledge about the task at hand, the users role, the state of the processes that the user is involved in and other relevant constraints. Context-awareness was originially oriented towards physical context [12] of mobile user interfaces, but was early identified as an important property of clincial systems in general [13]. In previous projects, we have observed features of context and behaviour and interpreted them in terms of requirements to mobile clinical information [14, 1, 3].

3.3 Decision support

Decision support is in a sense only a specialization of process- and collaboration support [15]. However, so far it is still a major challenge just to integrate decision support into clinical information systems [16] , which is a prerequisite for further development [2]. Based on an extensive review of literature, Wright and Sittig [17] describes the evolution of decision support systems, and discusses several possible features. They introduce four phases of integration of DSS and CIS (standalone, integrated into CIS, standards for knowledge sharing, service models), we see our work as presented as first attempt to define a fifth phase: Decision support integrated with pervasive, process-aware interfaces to clinical information.

3.4 Guideline models

In a process-centric view of care, a guideline, or its instantiation as a plan, is subdivided into components of tasks and activities. Many guideline representations [18] or presentations [19] are quite explicit. However, this subdivision is made for the purpose of simplifying the coding or reasoning about the representation, and not neccesarily relevant as a unit of work, ie. a session. We regard it as a major challenge to clarify the relation between a component of a guideline or plan, and the proper subdivision of work as seen from a clinicians point of

view. Earlier, we have discussed grounding of guidelines [20] and representation of collaborative care trajectories [21, 22]. We hope that the ideas presented here will be a fundament for capturing the relationship between process as prescribed and process as enacted.

4 Towards attributes of session-aware systems

For our purposes, we define a *session* as a time-delimited dialogue between two or more actors, one of which is a computer system, and which is representable in a computer system. The session, as supported by a computer system, is a sharable, referable, persistent representation of interaction with a context, a start, a (preliminary) end and an ordered set of explicit messages, commands or computer directives (eg. scrolling and clicking).

In the scenario, we have alluded to some of the properties for a system supporting sessions. This section elaborates more upon these attributes of a session before we propose a set of requirements to session-aware systems.

interruptable: A resident is conducting a meeting with a patient and is made aware of a different, more urgent task somewhere else on the ward. This new, imminent task forces him to postpone what was left of the previous task, often with an intention to resume the work as soon as the more urgent task has been completed. As part of the first task, the resident may have queried knowledge sources, read or edited the patient's medical record. Being able to freeze the GUI of the medical record, store the session and resume the interaction with the medical record as soon as the resident resumes the patient meeting would mean a huge advantage.

Multidevice: During clinical work, ward physicians and nurses frequently run into problems they not are capable of solving alone. In this situation, a reasonable strategy may be to postpone solving the problem until it can be presented and discussed with more experienced colleagues at a staff meeting. In practice this means to pause the problem solving task and resume it at a different time and location and on a different device like electronic blackboards and projectors. The problem owner should be able to move her session from her own device to a device suited for simultaneous viewing by those who attend the meeting, with presentation and level of detail (and patient privacy) adjusted as needed.

Transferable: In some situations, a nurse or physician working at the ward may not be able to complete a given clinical task forcing the hand over of the responsibility of the task to another actor, without actually handing over the tool, system interface or anything physical. This is also the situation when the expected duration of a task exceeds the duration of a shift. Ideally, the second actor should be able to continue where the first actor stopped without having to spend too much time on catching up on his task.

Conditionally pausable: A clinical task must be brought to a pause and postponed until some planned future event has occurred or until a defined period

of time has passed. This may for instance be the case when the task is to evaluate the outcome of a surgical procedure or some other therapeutic event: Some time must pass before the clinical parameters can be collected and evaluated. To maximize the learning from experience, the person who conducted the therapeutic procedure also should undertake the evaluation. In this situation he should be able to postpone tasks to himself.

Abstractable parametrisable: The nurses in a ward quickly learn to set up a session for a specific task, and then switch from one patient to another in the same session. Different wards and groups of professionals set up templates, and store them in a common library for different types of patiens, as for sessions that can be instantiated for different situations and patients. An example session is "write discharge summary for coronary ward patient to be treated in nursing home". Specific sessions correspond to parts of the procedures and guidelines used in the hospital. When developing new guidelines, the different responsible groups take care to develop template sessions where needed.

As a first step towards describing requirements for systems supporting sessions, we present what we believe are the main attributes of functionality in table 1:

5 Discussion

We have barely touched upon the rich research within CSCW, health sociology, DSS and HCI that are relevant for an analysis and implementation of session-aware systems. However, we believe that our main problem is rather banal: Our operating systems of choice, and system architectures, do simply not support a notion of portable, persistent, referable, suspendable, parametrizable session as a first class object. Persistent workspaces were commonplace in the Lisp-machines of the 80's and Unix-systems with X-windows. We are stuck with the sad idea of "Personal Computers" and physical devices, while we actually need "Shared Sessions", and higher abstractions of computer-supported work.

A different challenge is of organizational nature: Session-based systems may lead to fragmentation of responsibility and focus. Only high-fidelity experiments and simulations can uncover if that will be a problem.

Our research group is actively working on collaboration as communication support. Communication processes are the core of clinical work, but much of the communication is terse and implicit. Clinical information systems that take part in, and support, this communication must also know the context. The finding that a major part of the context was "fluid", led to a refined understanding of context as sessions. This paper has only presented a system-perspective of sessions.

Table 1. Attributes of functionality for session-supporting systems

Attribute	**A system should support sessions that** ...
Persistent	independent of physical user interface, device, and place, and can be stored, named, and referred globally.
Referrable	are possible to retrieve by name, with names in a naming-system/space.
Autonomous	can be active and running, regardless of number of watchers/users.
Separable	are possible to separate into presentation, interaction and contents
Continuous	have interaction history for each presentation that can be reviewed and replayed. The content is versioned.
Multi-user	can be viewed through different user interfaces, on various platforms, simultaneously
Access control	separate access control for content, interface(s), and interaction.
Fixable	have a momentarily instance, which can be used for documentation purposes.
Abstractable	allow partial abstraction on parts of a presentation.
Parametrized	are possible to store as templates of interaction, or presentations that can be given content with declarative queries.
Declarative	can be represented declaratively by query and interface specification language that makes them possible to generate and program.
Composite	can be grouped, and refer to or contain other sessions.
Comparable and searchable	can be related by identity and similarity. Ie. a representation of sessions should support queries like "find session Y among my sessions" and "Find sessions similar to the current for other patients in the ward."

6 Future Work

We are currently working on the POCMAP[2]-project, that aims to demonstrate session-support and plan-aware interfaces for mobile devices. So far, we have:

- performed extensive observations of collaboration and communication in real-life care sessions.
- run workshops and plays with clinicians and mock-up paper prototypes.
- started building prototype process support based on various guideline execution engines.
- played with Bardrams ABC-framework [23] with an aim to implement separation of interface/content and support for naming and reference.

[2] Point Of Care Multi-Aware Pilot

Within a year, we hope that we will be able do demonstrate and evaluate prototype clinical information systems that can support sessions in some of the ways we have described.

References

[1] I. D. Sørby and Ø. Nytrø. Towards a tomographic framework for structured observation of communicative behaviour in hospital wards. In P. Sawyer, B. Paech, and P. Heymans, editors, *REFSQ 2007 - Proceedings of the international working conference on Requirements Engineering - Foundations for Software Quality*, volume 4542 of *Lecture Notes in Computer Science*, pages 262–276. Springer-Verlag, Berlin, 2007.

[2] S Tu, MA Musen, and R Shankar. Modeling guidelines for integration into clinical workflow. In M. Fieschi, E. Coiera, and Y.-C. J. Li, editors, *MEDINFO 2004 - Proceedings of the 11th World Congress on Medical Informatics*, volume 107 of *Studies in Health Technology and Informatics*, pages 174–178. IOS Press, Amsterdam, 2004.

[3] Y. Dahl, I. D. Sørby, and Ø. Nytrø. Context in care - requirements for mobile context-aware patient charts. In M. Fieschi, E. Coiera, and Y.-C. J. Li, editors, *MEDINFO 2004 - Proceedings of the 11th World Congress on Medical Informatics*, volume 107 of *Studies in Health Technology and Informatics*. IOS Press, Amsterdam, 2004.

[4] International Telecommunication Union. X.225: Open systems interconnection - connection-oriented session protocol specification, 1995.

[5] D. Garfinkel, B.C. Welti, and T.W. Yip. HP SharedX: A Tool for Real-Time Collaboration. *Hewlett-Packard Journal*, 45(2):23–36, 1994.

[6] Alan Dearle. Toward ubiquitous environments for mobile users. *IEEE Internet Computing*, 02(1):22–32, 1998.

[7] M. Satyanarayanan et al. Pervasive computing: vision and challenges. *IEEE Personal Communications*, 8(4):10–17, 2001.

[8] B. Johanson, A. Fox, and T. Winograd. The Interactive Workspaces Project: Experiences with Ubiquitous Computing Rooms. *IEEE Pervasive Computing*, 01(2):67–74, 2002.

[9] Jakob E. Bardram, Jonathan Bunde-Pedersen, and Mads Soegaard. Support for activity-based computing in a personal computing operating system. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 211–220. ACM Press, 2006.

[10] Jakob E. Bardram and Claus Bossen. Mobility work: The spatial dimension of collaboration at a hospital. *Computer Supported Cooperative Work (CSCW)*, 14(2):131–160, 2005.

[11] Claus Bossen and Jens Bæk Jørgensen. Context-descriptive prototypes and their application to medicine administration. In *Proceedings of the 2004 conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 297–306, Cambridge, MA, USA, 2004. ACM Press.

[12] Bill N. Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pages 85–90, 1994.

[13] DF Sittig, JM Teich, JA Yungton, and HC Chueh. Preserving context in a multi-tasking clinical environment: A pilot implementation. In *AMIA Fall Symp*, pages 784–788, 1997.

[14] I. D. Sørby and Ø. Nytrø. A study on clinicians' information systems usage in drug administering and prescription situations. *Accepted for Poster presentation at Medinfo 2007 (The 12th World Congress on Health (Medical) Informatics)*, 2007.

[15] R. Lenz and M. Reichert. IT support for healthcare processes–premises, challenges, perspectives. *Data & Knowledge Engineering*, 61(1):39–58, 2007.

[16] H. van der Linden, S. Diepen, G. Boers, H. Tange, and J. Talmon. Towards a Generic Connection of EHR and DSS. In *Connecting Medical Informatics And Bio-informatics: Proceedings of MIE2005*, volume 116 of *Stud Health Technol Inform*, pages 211–216. IOS Press, 2005.

[17] A. Wright and D.F. Sittig. A four-phase model of the evolution of clinical decision support architectures. *International Journal of Medical Informatics*, 2008.

[18] E. Shalom and Y. Shahar. A Graphical Framework for Specification of Clinical Guidelines at Multiple Representation Levels. *AMIA Annual Symposium Proceedings*, 2005:679, 2005.

[19] R. Kosara and S. Miksch. Metaphors of movement: a visualization and user interface for time-oriented, skeletal plans. *Artificial Intelligence In Medicine*, 22(2):111–131, 2001.

[20] Inger Dybdahl Sørby, Øystein Nytrø, and Thomas Brox Røst. Empirical grounding of guideline implementation in cooperative clinical care situations. In A. ten Teije, S. Miksch, and P. Lukas, editors, *AI Techniques in Healthcare: Evidence-based Guidelines and Protocols (workshop at ECAI 2006)*, pages 89–94, Riva del Garda, Italy, 2006.

[21] I. D. Sørby, L. Melby, and Ø. Nytrø. Characterizing cooperation in the ward: framework for producing requirements to mobile electronic healthcare records. *Int. Journal of Healthcare Technology and Management*, 7(6):506–521, 2006.

[22] I. D. Sørby and Ø. Nytrø. Analysis of communicative behaviour: Profiling roles and activities. In Johanna I. Westbrook, Enrico W. Coiera, Joanne L. Callen, and Jos Aarts, editors, *Proceedings of the 3rd International Conference on Information Technology in Health Care: Socio-technical Approaches*, volume 130 of *Studies in Health Technology and Informatics*, pages 111–120. IOS Press, Sydney, Australia, 2007.

[23] Jakob E. Bardram and Henrik B. Christensen. Pervasive computing support for hospitals: An overview of the activity-based computing project. *IEEE Pervasive Computing*, 6(1):44–51, 2007.