Bruno Siqueira Campos Mendonça Vilar

"Context driven Workflow Adaptation applied to Healthcare Planning"

"*Adaptação de Workflows dirigida por Contexto aplicada ao Planejamento de Saúde*"

CAMPINAS
2014

UNICAMP
University of Campinas
Institute of Computing

Universidade Estadual de Campinas
Instituto de Computação

Bruno Siqueira Campos Mendonça Vilar

"Context driven Workflow Adaptation applied to Healthcare Planning"

Supervisor: *Orientador(a):* **Prof^{a}. Dr^{a}. Claudia Maria Bauzer Medeiros**

Co-Supervisor: *Co-orientador(a):* **Prof. Dr. André Santanchè**

"Adaptação de Workflows dirigida por Contexto aplicada ao Planejamento de Saúde"

PhD Thesis presented to the Post Graduate Program of the Institute of Computing of the University of Campinas to obtain a PhD degree in Computer Science.

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Computação da Universidade Estadual de Campinas para obtenção do título de Doutor em Ciência da Computação.

THIS VOLUME CORRESPONDS TO THE FINAL VERSION OF THE THESIS DEFENDED BY BRUNO SIQUEIRA CAMPOS MENDONÇA VILAR, UNDER THE SUPERVISION OF PROF^{A}. DR^{A}. CLAUDIA MARIA BAUZER MEDEIROS.

Este exemplar corresponde à versão final da Tese defendida por Bruno Siqueira Campos Mendonça Vilar, sob orientação de Prof^{a}. Dr^{a}. Claudia Maria Bauzer Medeiros.

Supervisor's signature / *Assinatura do Orientador(a)*

CAMPINAS
2014

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

M467c Vilar, Bruno Siqueira Campos Mendonça, 1982-
Context driven workflow adaptation applied to healthcare planning / Bruno Siqueira Campos Mendonça Vilar. – Campinas, SP : [s.n.], 2014.

Orientador: Claudia Maria Bauzer Medeiros.
Coorientador: André Santanchè.
Tese (doutorado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Sistemas de gestão de fluxo de trabalho. 2. Fluxo de trabalho - Administração - Programas de computador. 3. Fluxo de trabalho - Processamento de dados. 4. Hospitais - Gerenciamento de recursos de informação. 5. Saúde - Planejamento - Processamento de dados. 6. Software - Desenvolvimento. I. Medeiros, Claudia Maria Bauzer,1954-. II. Santanchè, André,1968-. III. Universidade Estadual de Campinas. Instituto de Computação. IV. Título.

Informações para Biblioteca Digital

Título em outro idioma: Adaptação de workflows dirigida por contexto aplicada ao planejamento de saúde
Palavras-chave em inglês:
Workflow management systems
Workflow - Management - Computer programs
Workflow - Data processing
Hospitals - Information resource management
Health planning - Data processing
Computer software - Development
Área de concentração: Ciência da Computação
Titulação: Doutor em Ciência da Computação
Banca examinadora:
Claudia Maria Bauzer Medeiros [Orientador]
Cecília Mary Fischer Rubira
Ana Estela Antunes da Silva
Fátima de Lourdes dos Santos Nunes Marques
Lucinéia Heloisa Thom
Data de defesa: 14-04-2014
Programa de Pós-Graduação: Ciência da Computação

TERMO DE APROVAÇÃO

Defesa de Tese de Doutorado em Ciência da Computação, apresentada pelo(a) Doutorando(a) **Bruno Siqueira Campos Mendonça Vilar**, aprovado(a) em **14/4/2014**, pela Banca examinadora composta pelos professores doutores:

Prof(ª). Dr(ª). Fátima de Lourdes dos Santos Nunes Marques
Titular

Prof(ª). Dr(ª). Lucinéia Heloisa Thom
Titular

Prof(ª). Dr(ª). Cecília Mary Fischer Rubira
Titular

Prof(ª). Dr(ª). Ana Estela Antunes da Silva
Titular

Prof(ª). Dr(ª). Claudia Maria Bauzer Medeiros
Presidente

Institute of Computing /*Instituto de Computação*
University of Campinas /*Universidade Estadual de Campinas*

Context driven Workflow Adaptation applied to Healthcare Planning

Bruno Siqueira Campos Mendonça Vilar[1]

April 14, 2014

Examiner Board/*Banca Examinadora*:

- Prof^{a}. Dr^{a}. Claudia Maria Bauzer Medeiros (Supervisor/*Orientadora*)
- Prof^{a}. Dr^{a}. Cecília Mary Fischer Rubira
  Institute of Computing - UNICAMP
- Prof^{a}. Dr^{a}. Ana Estela Antunes da Silva
  Faculty of Technology - UNICAMP
- Prof^{a}. Dr^{a}. Fátima de Lourdes dos Santos Nunes Marques
  Escola de Artes, Ciências e Humanidades - USP
- Prof^{a}. Dr^{a}. Lucinéia Heloisa Thom
  Institute of Informatics - UFRGS

[1]Financial support: Work partially financed by FAPESP (grant 2011/17196-0), FAPESP/Cepid in Computational Engineering and Sciences (2013/08293-7), the Microsoft Research FAPESP Virtual Institute (NavScales project), CNPq (MuZOO Project), FAPESP-PRONEX(eScience project), INCT in Web Science, and individual grants from CNPq and CAPES.

Abstract

Workflow Management Systems (WfMS) are used to manage the execution of processes, improving efficiency and efficacy of the procedure in use. The driving forces behind the adoption and development of WfMSs are business and scientific applications. Associated research efforts resulted in consolidated mechanisms, consensual protocols and standards. In particular, a scientific WfMS helps scientists to specify and run distributed experiments. It provides several features that support activities within an experimental environment, such as providing flexibility to change workflow design and keeping provenance (and thus reproducibility) of experiments. On the other hand, barring a few research initiatives, WfMSs do not provide appropriate support to dynamic, context-based customization during run-time; on-the-fly adaptations usually require user intervention. This thesis is concerned with mending this gap, providing WfMSs with a context-aware mechanism to dynamically customize workflow execution. As a result, we designed and developed DynFlow – a software architecture that allows such a customization, applied to a specific domain: healthcare planning. This application domain was chosen because it is a very good example of context-sensitive customization. Indeed, healthcare procedures constantly undergo unexpected changes that may occur during a treatment, such as a patient's reaction to a medicine. To meet dynamic customization demands, healthcare planning research has developed semi-automated techniques to support fast changes of the careflow steps according to a patient's state and evolution. One such technique is Computer-Interpretable Guidelines (CIG), whose most prominent member is the Task-Network Model (TNM) – a rule based approach able to build on the fly a plan according to the context. Our research led us to conclude that CIGs do not support features required by health professionals, such as distributed execution, provenance and extensibility, which are available from WfMSs. In other words, CIGs and WfMSs have complementary characteristics, and both are directed towards execution of activities. Given the above facts, the main contributions of the thesis are the following: (a) the design and development of DynFlow, whose underlying model blends TNM characteristics with WfMS; (b) the characterization of the main advantages and disadvantages of CIG models and workflow models; and (c) the implementation of a prototype, based on ontologies, applied to nursing care. Ontologies are used as a solution to enable interoperability across distinct SWfMS internal representations, as well as to support distinct healthcare vocabularies and procedures.

Resumo

Sistemas de Gerenciamento de Workflows (WfMS) são usados para gerenciar a execução de processos, melhorando eficiência e eficácia de procedimentos em uso. As forças motrizes por trás da adoção e do desenvolvimento de um WfMS são aplicações científicas e de negócios. Esforços conjuntos de ambos resultaram em mecanismos consolidados, além de padrões e protocolos consensuais. Em particular, um WfMS científico (SWfMS – *Scientific WfMS*) auxilia cientistas a especificar e executar experimentos distribuídos. Ele fornece diferentes recursos que suportam atividades em um ambiente experimental, como prover flexibilidade para mudar o projeto de workflow, manter a proveniência e suportar reproducibilidade de experimentos. Por outro lado, apesar de poucas iniciativas de pesquisa, WfMSs não fornecem suporte apropriado à personalização dinâmica e baseada em contexto durante a execução; adaptações em tempo de execução normalmente requerem intervenção do usuário. Esta tese se concentra em superar essa deficiência, fornecendo a WfMSs um mecanismo de ciente do contexto para personalizar a execução de workflows. Como resultado, foi projetado e desenvolvido o DynFlow – uma arquitetura de software que permite tal personalização aplicada a um domínio: planejamento de saúde. Este domínio foi escolhido por ser um ótimo exemplo de personalização sensível ao contexto. Procedimentos de saúde constantemente sofrem mudanças que podem ocorrer durante um tratamento, como a reação de um paciente a um medicamento. Para suprir a demanda, a pesquisa em planejamento de saúde desenvolveu técnicas semi-automáticas para suportar mudanças rápidas dos passos de fluxos de tratamento, de acordo com o estado e a evolução do paciente. Uma dessas técnicas é *Computer-Interpretable Guidelines* (CIG), cujo membro mais proeminente é *Task-Network Model* (TNM) – uma abordagem baseada em regras capaz de construir um plano em tempo de execução. Nossa pesquisa nos levou a concluir que CIGs não suportam características necessárias por profissionais de saúde, como proveniência e extensibilidade, disponíveis em WfMSs. Em outras palavras, CIGs e WfMSs têm características complementares e são direcionadas à execução de atividades. Considerando os fatos citados, as principais contribuições desta tese são: (a) especificação e desenvolvimento do DynFlow, cujo modelo associa características de TNMs e WfMS; (b) caracterização das principais vantagens e desvantagens de modelos CIGs e WfMSs; (c) implementação de um protótipo, baseado em ontologias e aplicadas ao domínio da saúde e enfermagem.

Acknowledgements

This thesis is the result of a long process in which many people participated – either on a daily contact or on a brief moment, but with some gesture that made the difference. Although these people are in a higher level than this work, I would like to thank and dedicate it to each of them. I will try to list some of these people on the following lines.

My advisor, Claudia Medeiros who accepted me as student, guiding me with hard lessons and wise advices that were essential for my growth. My co-advisor, André Santanchè, who contributed with other hard lessons and also for the great wittiness.

My family. Especially my mother, Ulemá, for all support regarding my decisions. Also to my father Tércio, and (all) my sisters: Ana Paula, Beatriz, Isabela, Juliana and Vivianne. I would also like to thank Bibita, who take care of me in several occasions, and Tulio (*in memoriam*).

Professors from CEULP/ULBRA: My advisors Fabiano and Parcilene, who inspired me in many ways. Also, professors Cristina, Fernando, Jackson, Madianita, Ricardo and Thereza.

Friends from LIS – besides all photos and hugs. Alan, Alessandra, Andréia, Carla, Luis Theodoro, Luiz Celso, Daniel, Eduardo, Evandro, Gabriel, Gilberto, Ivelize, Ivo, Jaudete, Jacqueline, Joana, Jordi, Lucas, Matheus, Nielsen, Renato and Senra.

Friends from IC-UNICAMP and Campinas. Bruno Azevedo, Carlos, Crístian, Elaine, Erikson, Fábio, Fabiana, Gau, Geraldo, Gilberto, Glauber, Jefersson, Lin, Otávio, Priscila, Rachid, Roberto, Tripodi. Also to professors Ariadne, Rodolfo, Rubira, Tomazs, and Torres, as well as other members of IC, including Daniel Capeleto, Natália and Rosa.

Friends from Palmas, Tocantins. Álvaro, Carlos André, Danielle, Edeilson, Emileny, Elizabeth, Fabrício Correia, Fábio, Francine, Hamistelie, Helena, Ilana, Ishibashi, Jacqueline, Katia, Karol, d. Nilva, Ricardo, Tatiana and Warley.

Friends that may be contextualized in other groups, but share some special projects and goals: Ricardo Panaggio, Marcela Lima, Marcus Saad and Raniere Gaia.

Funding Agencies: FAPESP (grant 2011/17196-0) CNPq and CAPES. Also Maria Helena, Neurilene and other healthcare providers from the Unicamp and USP, who provide us support.

Especially, I would like to thank Thatiane, a longtime friend which accumulates more roles in my life and becomes more important to me everyday.

Contents

Glossary

ADT = Augmented Decision Tables
API = Application Programmer Interface
AR = Adaptation Rules
BMI = Body Mass Index
BPEL = Business Process Execution Language
CAExt = Context Adaptation Extension
CIG = Computer-Interpretable Guidelines
CRec = Context Record
CPG = Clinical Practical Guidelines
CPO = Clinical Pathway Ontology
DAML = DARPA Agent Markup Language
DB = Database
EHR = Electronic Health Record
FIO2 = Fraction of Inspired Oxygen
HL7 = Health Level Seven
IT = Information Technology
I-RDS = Infants' Respiratory Distress Syndrome
KAR = Kepler Archive
MLM = Medical Logic Modules
NANDA = North American Nursing Diagnosis Association
NIC = Nursing Interventions Classification
NOC = Nursing Outcomes Classification
ODP = Open Directory Project
OIL = Ontology Inference Layer
OWL = Ontology Web Language
OWL-S = Ontology Web Language for Services
pCO2 = Pressures of Carbon Dioxide
PEEP = Positive End-Expiratory Pressure
PiP = Peak inspiratory Pressure

PROCEnf = Nursing Process Electronic Documentation System of the University of São Paulo
RDF = Resource Description Framework
SPARQL = SPARQL Protocol and RDF Query Language
SOAP = Simple Object Access Protocol
SWfMS = Scientific Workflow Management System
SWRL = Semantic Web Rule Language
SVN = Subversion
TcSaO2 = Transcutaneous Oxygen Saturation
TNM = Task-Network Model
UML = Unified Modeling Language
URI = Uniform Resouce Identifier
URN = Uniform Resource Name
uWLD = Ubiquitous Workflow Description Language
WfMC = Workflow Management Coalition
WfMS = Workflow Management System
XML = Extensible Markup Language
XPATH = XML Path Language
XSLT = Extensible Stylesheet Language Transformations
YAWL = Yet Another Workflow Language

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview of the problem

The need to model, analyze and automate processes to gain productivity is not recent, especially in a business context [11]. This need originated the creation of Workflow Management Systems (WfMS), now a consolidated research domain, with consensual protocols and standards, notably those proposed by the Workflow Management Coalition (WfMC) [37]. Since the need to improve efficiency and efficacy is not exclusive to business, the use of similar mechanisms is also observed in other domains, in particular to support scientific research practices.

A research lab is characterized by specification and execution of experiments. Scientific experiments require repetition cycles of execution-validation-modification with different trial and error activities, until success. This originated a specific kind of workflow: scientific workflows. A scientific workflow represents the process chaining that transforms data aiming at an experimentation by simulation [54].

The specific needs and high demands of a workflow mechanism for scientific experiments was a second driving force of workflows and originated a specialized version of WfMS: Scientific Workflow Management Systems (SWfMS) [71]. This kind of software provides five important features [45, 26, 71]:

- Graphical interface to manage computational resources: WfMSs (including SWfMS) provide a graphical user interface that allows to click and drag visual representations of computational tools that are automatically executed within a workflow. This helps users to manage computational resources, without need to know systems programming, making it easier for scientists and users to automate experiments and tasks.

- Flexibility to change workflow design and parameters: a workflow can be changed to add a new step on an experiment, decompose steps, or have parameters changed to validate a new hypothesis, again using the interface.

- Provenance of experiments: most WfMS support activity tracking, and history of executions and changes, thereby allowing scientists to maintain the provenance of data produced, and processes data went through.

- Reproducibility of experiments: using provenance information, it is possible to re-execute an experiment to check the results obtained and validate it.

The popularization of scientific workflows brought new challenges. For instance, "with increasing complexity it becomes more and more unfeasible to optimize scientific workflows by trial and error" [35]. Complexity also reduces some of the benefits of the highlighted features. Workflows are designed in advance, and their specification will be followed by a WfMS. Even though there are approaches to promote changes in the plan according to the context [74, 20, 59, 12, 63], they will be small controlled changes, which maintain the main workflow structure. However, the dynamicity of some domains, like healthcare, will often require major changes and adaptations in the plan. The workflow model and consequently WfMSs were not conceived to support such dynamic major adaptations. Moreover, the flexibility to change the workflow design and its parameters is lost when they are complex, because it is not simple to understand all consequences of a change made on a workflow, its activities and all input and output parameters involved.

A direction to increase flexibility is to allow workflows to promote on-the-fly adaptation according to the context. According to [28], context is "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves". An adaptation mechanism can be developed allowing workflow systems to have access to the data used in all processes and the way they processes are related. This is not a default scenario in workflow systems because the data involved is usually hidden behind task implementation. Accessing all information related to the task allows workflow systems to understand the execution panorama and thus adapt to new situations.

1.2 Scientific Workflow Customization

Throughout this text, the terms "workflow" and "workflow system" will refer to scientific workflows and systems, unless otherwise stated. From time to time this will be restated, to help the reader.

Scientific Workflow environments allow, in different flexibility levels, contextualization and adaptation of workflows, considered representation of the executable experiment. Often, these workflows are available in repositories, to be re-executed. Such workflows are also used to document an experiment and to determine the provenance of data and/or processes.

However, frequently a workflow must be adapted according to the user (scientist) “context”: for example, changing parameters or modules to invoke. Another kind of possible adaptation is the modification of the workflow itself, changing one or more activities that compose it, or adding new branches. All these possibilities, which are necessary in a scientific experimental environment, depend on the user’s knowledge – not only about the target domain, but also about the availability of data, execution environment and others. Workflow specification environments should therefore aid scientists in such kinds of adaptation.

Ideally, a SWfMS should support on-the-fly context sensitive customization, considering two types of adaptation:

- keeping the same workflow specification, but changing the execution conditions, for example, changing paramaters values or modules to invoke;
- modifing the workflow, for instance, adding/removing activities or changing control structures.

Let us consider a basic workflow, such as the one portrayed in Figure 1.1. Here, we differentiate between an abstract workflow (the specification) – top part of the figure – and its executable version, which results from the activation of concrete software modules – bottom of the figure – related to the execution of activities. Figure 1.1 shows an abstract workflow W, with three activities (A1, A2, A3) and data input from a repository R. Software modules M1, M2 and M3 are specified to be invoked when executing W.

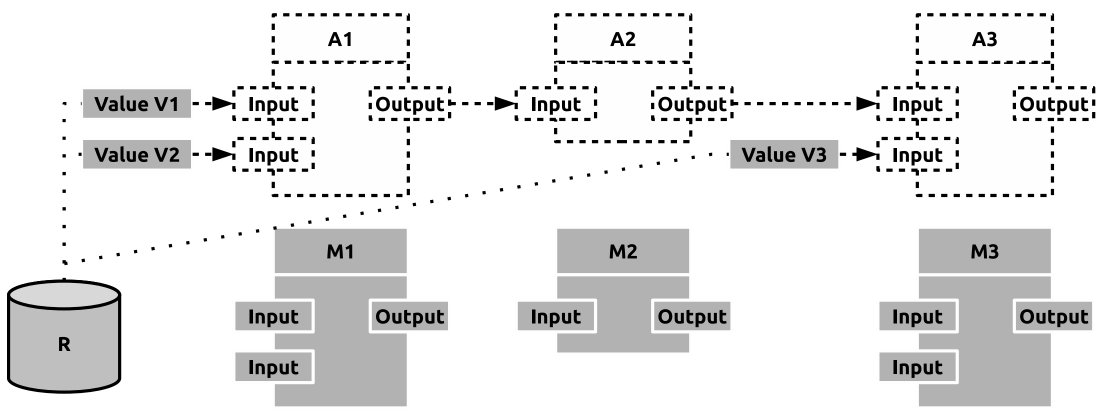

Figure 1.1: Abstract workflow W, modules M1, M2 and M3 are invoked during execution.

As Figure 1.2 shows, during runtime customization, the SWfMS, “becomes aware” that M5 should be invoked instead of M2, and this modification is performed on-the-fly. Another possible modification might involve changing the input repository, or activity parameters. Figure 1.3 portraits that scenario.

Exemplifying the second kind of customization, the workflow structure is changed. Figure 1.4 shows this change – The SWfMS adds a new activity because of the execution context.

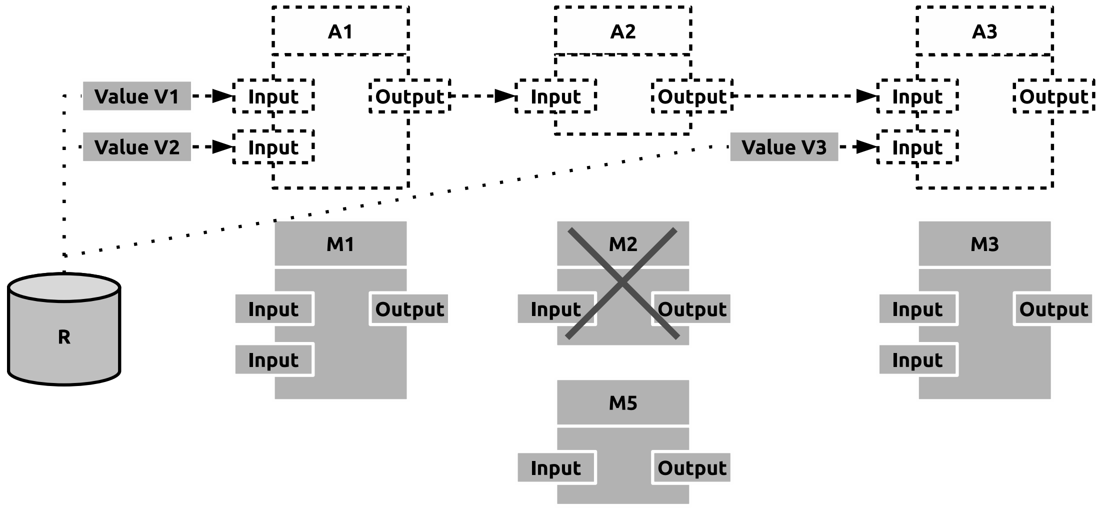

Figure 1.2: Executable workflow for W. The SWfMS replaced module M2 by M5 due to context customization.

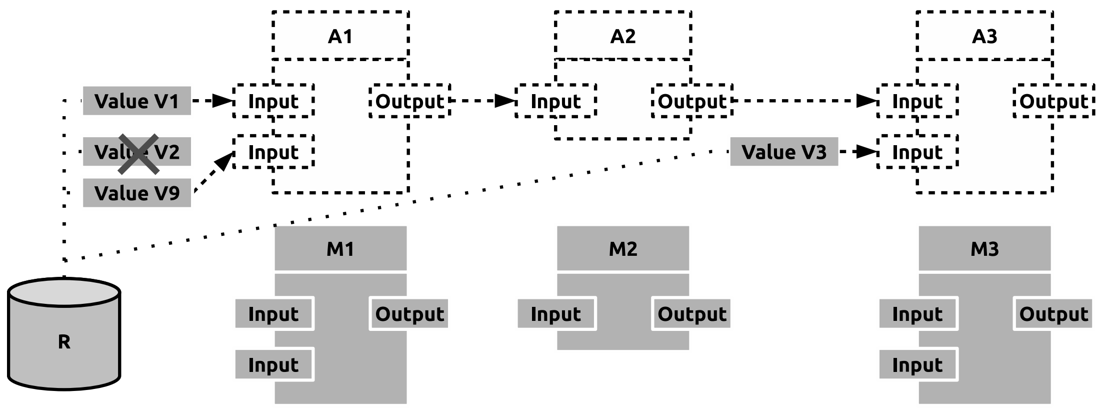

Figure 1.3: Customized workflow, with change of parameter value (V9) invoking module M1 on activity A1.

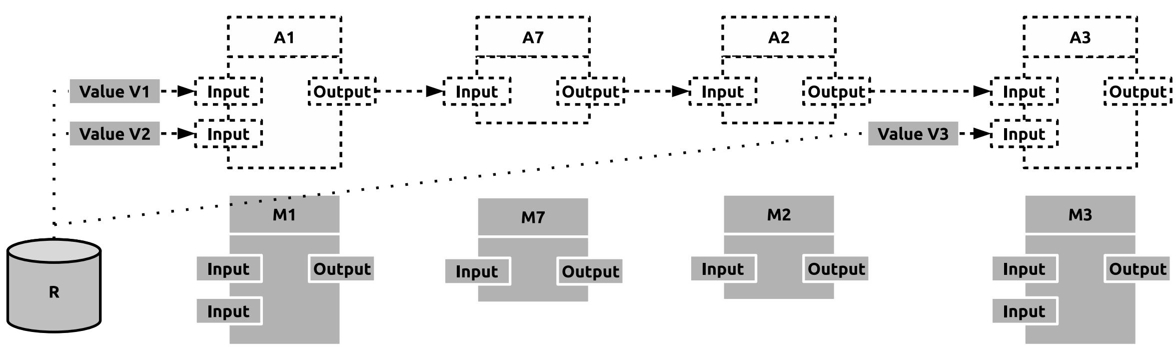

Figure 1.4: Customized workflow, with addition of other activity (A7) invoking module on M7.

There have been several research efforts directed to customization and adaptation of workflows and processes. Similar to workflow patterns initiative [77], Russell et al. [62] specified changes related to workflow data, resulting in workflow data patterns. According to the authors, there are four distinct groups of patterns related to data:

- Visibility: controls the context in which values can be accessed under different criteria (e.g. by instance of a task or under specific scope);
- Interaction: defines how data elements can be passed between elements of an workflow process (e.g. communication between instances of different tasks or from a task and a sub-workflow task).
- Transfer: similar to interaction, this kind of pattern deals with the way data are passed between workflow components, specifying aspects like transfer by value of reference and concurrency restrictions.
- Routing based on data: indicates how the data can influence the flow of processes. For instance, specify pre and post conditions of processes parameters or a trigger to initiate a task.

Other groups specify similar approaches to deal with data flow, such as workflow change patterns [1]. As will be seen on Chapters 3 and 4, we adapted some of those concepts that might indirectly be related to such patterns, applied to the process of nursing care anamnesis, which involves filling in forms with results of the patient's interrogations:

- Change of execution conditions – data collected during the process may impact in changing parameters of other actions.
- Change of workflow specifications – the value of a given parameter (e.g., patient's age) may impact in the workflow specification: several activities are eliminated, and others are prioritized, changing execution order.

1.3 Organization and Contributions

This thesis concentrates in attacking some of the challenges of enhancing scientific workflow systems so that workflows can be dynamically adapted, on the fly, according to the running context. This adaptation is data-driven. Our work is validated for applications in healthcare planning. This is an area in which it is necessary to plan actions that are suitable to a given condition of a patient – e.g. provide a treatment to a diagnosis. However, at any moment a new situation may occur, as a reaction to a medicine or the manifestation of a new symptom. This is therefore a good scenario to test our work.

There are systems and models that are created to cope with the flexibility and dynamic scenarios of healthcare, especially Computer-Interpretable Guidelines (CIG) [56]. One of the most common CIG models found in the literature is the Task-Network Model (TNM) [34]. This model, unlike other approaches, provides full support for conceptualizing a multi-step guideline that unfolds over time [57] – and thus cope with evolution of a patient's state.

Comparing TNM to workflows led us to notice important distinctions that can be summarized making a parallel with two programming paradigms: procedural and logic approaches. A workflow specification is designed to behave like a procedural program, since its main flow of actions is previously defined, through an algorithmic set of actions. It can be defined by humans or by adaptive systems. A TNM specification can be compared to a rule based program, following a logics paradigm. Blocks of actions behave like rules, as they are omnipresent and will be triggered whenever their conditions are satisfied. Consequently, it is not possible to devise in the beginning the emergent behavior that results from the interaction of the rules (blocks of actions) with the (patient's) context.

Though CIG and TNM features allow flexible execution and efficient adaptation, there are some important requirements that remain incipient for healthcare planning. Examples are, the communication with external devices and programs, the provenance of data, simplicity of use and automation. Such features are, on the other hand, already provided by workflow systems. Given this situation, it is possible to identify that SWfMS and CIGs contain features that are complementary to each other. Based on these observations, we focused on the principle that, unlike related work, for healthcare planning, we would have to promote a synergy between workflow mechanisms and TNM. As result, our work creates a context-aware workflow mechanism inspired by healthcare planning techniques.

As Panzarasa and Stefanelli [56] highlight, "a critical challenge for any Workflow Management System (WfMS), in a real clinical setting, is its capability to respond effectively when exceptions occur. An exception can be defined as any deviation from the normal flow of activities, and it can arise from changes in resource availability, task requirements or task priority, and anomalous, but expected, effects of delivered care." WfMSs must moreover match other requirements to allow their use by healthcare professionals. They must support the rationale behind the procedures in the health context, allow the use of technical terms and expressions that describe concepts from the healthcare domain, use those concepts to guide care-flow and better respond to new information about patients.

There are initiatives that have investigated running workflows to automate healthcare tasks, e.g., [22], [50], and [65]. Workflows can also be used in health environments for running basic plans – e.g., to help control medication tasks. However, none of these initiatives considers incorporation of the flexibility and healthcare rationale provided by TNM features to dynamically adapt workflows to a context – and thus fail to appropriately customize healthcare procedures to a given situation. Our work is focused towards filling this gap. This thesis presents Dyn-

Flow – our software architecture that blends into scientific workflow systems the emergent and context-driven approach of care-flow systems. We have implemented DynFlow for the health-care domain, as a Web application, thereby simplifying its use by healthcare professionals, who do not need to install any specific software to use it.

Figure 1.5 gives an overview of the problems attacked in the thesis. SWfMS have the advantages of providing provenance, extensibility and robustness. CIGs, on the other hand, capture the healthcare rationale and provide flexibility to change the care-flow. The whole should be combined subject to context. As will be seen, this work brings TNMs into SWfMS.

The main contributions of this thesis are therefore:

- The characterization of the main advantages and disadvantages of healthcare planning models and workflow models according to a set of features.
- The specification of the DynFlow framework that blends strong features from both TNM and workflows, thereby supporting dynamic, on the fly, workflow customization with respect to a context.
- The implementation of a prototype for DynFlow, based on ontologies, applied to nursing care. Ontologies are used as a solution to provide interoperability across distinct SWfMS internal representations, as well as to support distinct healthcare vocabularies and procedures.

The originality of the contribution lies in the incorporation of TNMs into a SWfMS. Context-aware workflow adaptation (e.g. [18, 20, 39] does not take that kind of adaptation into consideration. Such adaptations are based, for instance, on rules and mostly consider business workflows [6, 66, 70, 61]. At the same time, Scientific WfMS do not easily provide dynamic adaptation.

This work is organized as a collection of papers, as follows. Chapter 2 is the paper "Towards Adapting Scientific Workflow Systems to Healthcare Planning" [78], published in the International Conference on Health Informatics. It presents our analysis comparing healthcare planning systems with workflow systems. This analysis explains the features based on a healthcare planning model (TNM) to provide flexibility for scientific workflows. The main contributions of this paper are:

- Characterization and comparison of WfMS and CIG systems under two perspectives: infrastructure and flexibility to change the flow of activities during execution.
- Review and comparative synthesis of research that uses context.
- Presentation of a preliminary architecture to adapt workflows to context.

Chapter 3 is the paper "Using healthcare planning features to drive scientific workflows", published in the 10th International Conference on Web Information Systems and Technologies

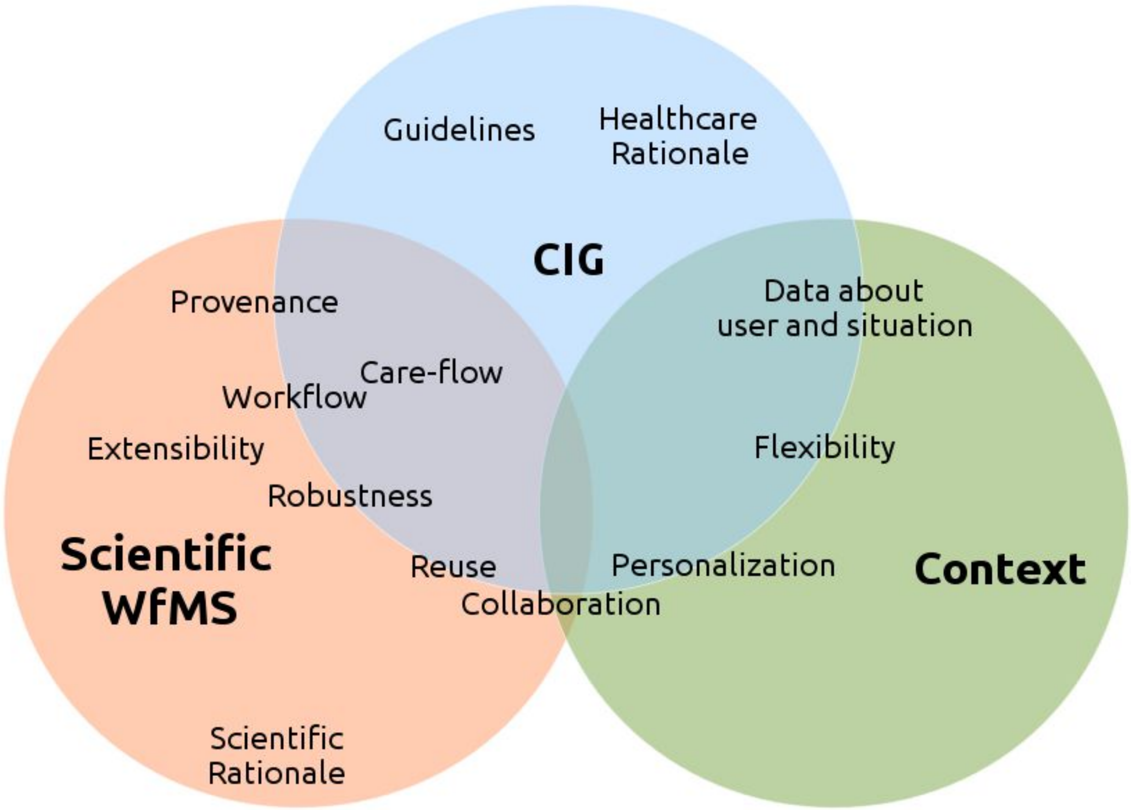

Figure 1.5: Overview of problems attacked and areas.

(WEBIST) [80]. This paper is geared towards the analysis of TNM and workflow models, allowing us to better identify the elements that allow healthcare planning systems to support unexpected changes under execution. As a result, we specify the DynFlow framework to allow dynamic workflow adaptation, applied to a nursing care case study. The contributions of this paper are:

- Specification of a framework that provides adaptation of workflows based on context.
- The presentation of a real life case study on nursing care.

Chapter 4 is the paper "DynFlow - a software architecture for dynamic adaptation of workflows for healthcare planning" submitted to a journal [79]. This paper extends and consolidates the results of the two previous chapters. It presents the complete specification of the framework and implementation details of our prototype. The contributions of the paper are:

- In-depth characterization of TNM features that provide flexibility to healthcare planning
- Refinement of the DynFlow framework to develop a context-aware workflow mechanism.

- Description of implementation details, in particular the context-based adaptation mechanism and the rationale behind the adoption of ontologies as the basis for interoperability and adaptation.

Chapter 5 presents our concluding remarks, as well as possible extensions. An overview of DynFlow modules and data flow within it is presented in Figure 1.6, reproduced from Chapter 4. It enables dynamic adaptation of workflows to a context as follows:

- A workflow is assigned to be executed on the SWfMS, which loads it and the user requests the SWfMS to start the executing workflow (arrow 1).

- Every time a new activity is performed, the *Context-Aware Extension* (CAExt) captures the workflow state and passes it to the *Semantic Mapper*. A workflow state contains activities, input and output parameters, and execution sequence. Next, the *Semantic Mapper* uses the workflow state to create an ontology instance that represents it – *Context Record*(CRec) (arrows 2, 3, and 4).

- The *Context Adaptation Engine* (CAEng) is a module that combines an Ontology, Adaptation Rules *AR*, and *CRec* to identify the adaptations recommended to the workflow. The Ontology embeds domain knowledge and concepts from the application/experiment being run. Rules define events or states to be captured and adaptations to perform based on context. Ontology, CRec and Rules are processed by an ontology reasoner that creates a new version of *CRec*: this version contains the state that the workflow should have (*CRec'*) to be adapted to the current situation.

- The *Semantic Mapper* receives *CRec'* and maps its data to the workflow representation, passing it back to the *CAExt* (arrows 5 and 6).

- The workflow on the SWfMS is updated by the *CAExt* according to the new workflow (arrow 7).

- Finally, the user sees the adapted version of the workflow (arrow 8).

We point out that by mapping the workflow representation into an ontology, DynFlow allows the use of different workflow systems, and for distinct domains, not being limited to healthcare. For each WfMS, it is necessary to personalize the Context-Aware Extension so that it acts as an interface to the remaining components. To use different domains, it is also necessary to create adaptation rules and ontologies that correspond to the related concepts and adaptations.

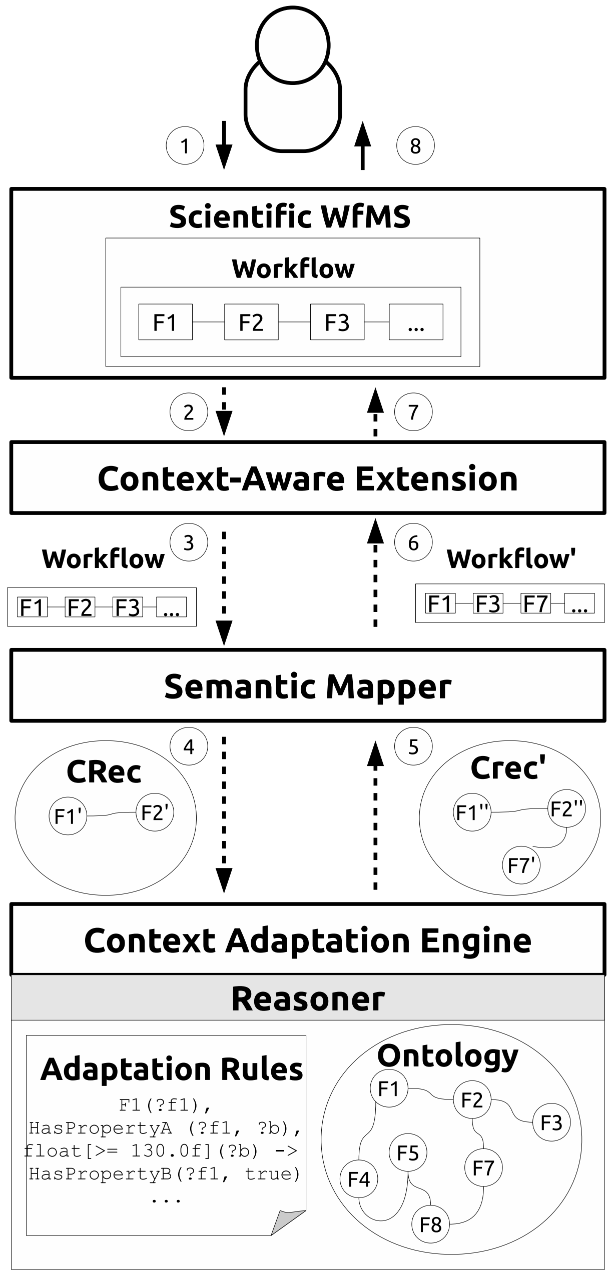

Figure 1.6: Overview of DynFlow for context-based workflow adaptation.

Chapter 2

Towards Adapting Scientific Workflow Systems to Healthcare Planning

2.1 Introduction

Healthcare facilities involve the management and coordination of healthcare providers, patients, and resources. There is a need for automated ways to monitor and integrate the flow of exams, nursing procedures and resources.

A common approach adopted to model healthcare processes are Computer-Interpretable Guidelines (CIGs), which implement guidelines in active computer-based decision support systems, able to monitor actions and observations of care providers and to provide guideline-based advice at the point of care [24]. CIG can be modelled as Task-network Model (TNM), which "decomposes guidelines into networks of tasks unfolding over time" [84]. A TNM can be seen as a hierarchical directed graph that specifies a flow of activities. Its enactment is often supported by artificial intelligence planning environments, in which a TNM is specified using some adaptation of a goal-based planning language.

Our key argument in this paper is that CIG systems, born in the healthcare context, embed the usual rationale applied in this context, tailored to the dynamic healthcare environment. Workflows, on the other hand, are robust tools, broadly tested and refined by the community for many domains. They are being increasingly adopted in hospitals – e.g., [60] – to support task automation, but are based on standard (business) workflow environments. Healthcare activities involve a dynamic scenario, in which professionals have to constantly interact with the tools, to register patient information, intervention plans, and desired outcomes, creating the need for flexible workflow management.

As Panzarasa and Stefanelli [56] highlight, "a critical challenge for any Workflow Management Systems (WfMS), in a real clinical setting, is its capability to respond effectively when exceptions occur. An exception can be defined as any deviation from the normal flow of activities, and it can arise from changes in resource availability, task requirements or task priority, and anomalous, but expected, effects of delivered care."

In fact, the generic modeling approach adopted by workflows, contrasts with the domain-specific formalisms (e.g., TNM) applied to model clinical practice guidelines. These formalisms are derived from the practical clinical activities and embed their rationale and approach to plan and manage activities.

There are research on adding flexibility to WfMS in a healthcare scenario, such as [22], [50], and [65]. There are, however, additional requirements to be fulfilled so that a WfMS can provide an adequate environment for practical clinic usage, such as preserve the systematization and work organization of clinicians, provide the traceability of actions, allow remote and collaborative work, among others.

Our solution to overcome these problems is based on two aspects: first, to adopt scientific workflow systems[1] ; second, to extend these systems with facilities for dynamic self-adaptation based on context. Our context-driven approach is a glue that enables to "think" workflows using a healthcare perspective.

Scientific workflow systems are normally adopted in research environments, to manage research activities, specification and execution of experiments. Given their event-driven characteristics, openness, flexibility, support to distributed collaborative work, and ability to handle exceptions, they are more suitable to health environments than business workflows.

This paper justifies this claim through analysis on the requirements of WfMS, created for scientific and business contexts, comparing their features with CIGs, applied to the healthcare domain. Our second ingredient – context-driven self-adaptation – is derived from the approach of planning and monitoring healthcare activities, as observed in TNM specifications.

The contributions of this paper are: (i) an outline of the architecture proposed with a case study (Section 4); (ii) a detailed analysis of the factors that led us this proposal, and comparative tables (Sections 2 and 3).

2.2 Related Work

2.2.1 Context

One of the aspects to be covered in this work is adaptation of workflows to a context. Defining the term "context" in an accurate or complete way is not a simple task: in the literature, it varies according to the perspective of who uses the term, and where it is used. [53] say that "it is very difficult to take into consideration all the contextual factors in one information retrieval system, so that researchers often define the context as certain factors (location for example)".

[1] As opposed to the business workflow systems adopted in hospitals.

[28] presents a generic definition of context, from a Computing perspective: "any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and application themselves". [73] define the term as: "the set of all information characterizing the entities relevant for a specific task in their relevant aspects".

The definitions of [14], [28] and [73] are generic with respect to the application. However, researchers more commonly require tailoring a concept to their needs, e.g., as in [5]. For them, "context describes the user current task, its changes over time and its states, i.e. we take into account the task which the user is undertaking when the information retrieval process occurs". [75], concerned with Intelligent Agents, defines context as "a distinguished (e.g., named) collection of possible world features that has predictive worth to the agent". To [7], it is "any static or dynamic client-, provider- or service-related information, which enables or enhances efficient communications among clients, providers and services".

The variation of the use of "context" in different fields and purposes results in diverse denominations. Table 2.1 summarizes our survey of the usages of the term. The table shows for each paper the application area, the domain of use and the representation of "context". This study also showed that diverse kinds of information can be used to specify "context", such as: those that identify a user's characteristics and preferences; or the location of an event and information about it, such as history, climatic conditions, legislation, service characteristics, domain, platform and others. This information is collected in different ways: sensors, logs of users' actions in systems, forms and others.

In spite of a wide space of variables to identify a context – eg. domain, service, location, identity, and device –, some can be highlighted. To [28], location, identity, time, and activity are the most important context variables to characterize the situation of an entity. According to the author, these kinds of contexts not only can answer questions such as "Who", "What", "When" and "Where", but also they can act as indexes to other contextual information sources. For instance, in all mobility and location-based studies, the most important variables are space (coordinates) and time.

2.2.2 Scientific Workflows

A workflow is the sequence of steps that are necessary to achieve a specific goal [10]. It allows to systematize a task in activities that, from their respective input resources, generate a certain result. Each activity can be composite or atomic. Workflows can be designed from an abstract specification that is refined gradually until it reaches an execution level – a concrete workflow [49]. An abstract specification helps to understand how workflow tasks are carried out, identify problematic spots (e.g., bottlenecks), and analyze changes that can be carried out in designing the workflow. Executable (or concrete) activities are those associated with some tool or service

Table 2.1: *Characteristics of work using the notion of context*

Work	Computing-related aspects	Application domain	Context Representation in a computer
[28]	Context-Aware Computing	Generic	-
[53]	Context-Aware Environments	Pervasive Computing	Multiple Ontologies (OWL)
[14]	Evaluation Framework - Context Models	Generic	Generic: analyze and suggest models
[7]	Web Services Composition	Business	Ontology (OWL - light)
[43]	Information Retrieval	Generic	Relational DB
[5]	Information Retrieval	Generic	Tasks represented as UML State Diagram. Terms associated to ontologies (WordNET e ODP)
[16]	Adaptive Hypermedia	Generic/Validation: Tourism	Multi-ontology matrix and SWRL rules
[46] and [47]	Distributed Environment Information Sharing	Emergency Service Centres	Inference rules mapped on Workflows Simulates: state and consequence
[8]	Computer-Supported Cooperative Work	Hospital Clinicians	Not specified
[73]	Middleware	Generic	Ontology (represented in: OWL, DAML+OIL and F-Logic)
[18]	Workflows	Agriculture	RDF: restrictions. uWLD: context aware workflows language.
[75]	Intelligent Agents	Medical Diagnosis	C-Schemas (”frame-like”)

that processes or aids the obtention of results that serve as input to other activities.

Workflow management, coordination of processes and other functionalities are the responsibility of WfMSs. These systems orchestrate algorithms and computational processes, combining parallel and distributed processing, databases, artificial intelligence, among others, building a repository for experimentation through simulation [26].

WfMSs allow processes to be organized in different ways to meet requirements and processing needs. The main characteristic of a WfMS is process automation, involving the combination of activities performed by people and computers [37]. The role of these systems, however, is not limited only to the automation, but also allows to obtain process information in different levels of detail, besides systematically capturing provenance information of produced data [64].

In this paper, scientific workflows are adapted to healthcare. A scientific workflow can represent the process chaining that transforms data aiming at an experimentation by simulation[54]. These systems “enable researchers to collaboratively design, manage, and obtain results that involve hundreds of thousands of steps, access terabytes of data, and generate similar amounts of intermediate and final data products” [25]. Thus, scientists can focus on their research and not on computation management [26].

2.2.3 Computer-Interpretable Guidelines

"Clinical guidelines can be viewed as generic skeletal-plan schemata that represent clinical procedural knowledge and that are instantiated and refined dynamically by care providers over significant time periods" [68]. There are also specialized versions of the guidelines, e.g., Nursing Clinical Guidelines, which provide evidence-based instructions/recommendations about how to handle specific patient care issues [29].

Even though the guidelines represent clinicians' background about suggested ways to deal with health issues, [32] highlight that the clinicians' judgment may conflict with the general guidelines, so the treatment may differ from the one originally stated. This may occur because the guidelines are generic, and thus may not consider new knowledge about the treatment, or patient allergy to the medicine, etc.

To automate the process of guideline application, as well as avoid errors and improve the process, there is the study and development of Computer-Interpretable Guidelines (CIGs). There are several rich studies to provide support for clinicians, such as Clinical Decision Support Systems [31], Care Flow [50], Task Network Model [84], Clinical Pathway Management System [84], and Healthcare Information Management System [22]. To be effective, "these tools need to be simple to use, easily available, and work with different information systems in changing environments" [44]. For those characteristics, there is a need to: i) Preserve methodology and systematization already applied by clinicians; ii) Provide flexibility to adapt the guideline instance to the situation; iii) Dispose of open and extensible environment to add resources and better suit the solution to the problem.

Our work tries to comply with those requirements combining features from WfMS and CIG systems. Section 2.3 analyses characteristics from both systems. external resources as a workflow task. To add a task it is necessary to extend a specific Microsoft .NET class or use external tools to import webservices.

2.3 Comparing WfMS and CIG systems

Ideally, automated systems to support healthcare activities must comply with a variety of software requirements. They must be flexible, extensible by plug-ins, support different kinds of activities (services, languages, etc.), allow annotation of tasks, register provenance, provide access by a remote client, and support changes according to context variables. Also, they must maintain the nature of work of clinicians, be flexible to changes, responding to new information about patients. We analyzed different tools designed to automate and monitor activities with respect to these properties, as a necessary step to understand deficiencies and advantages of such tools. We present the results of this study in this section.

We separated the analyzed features in two groups, infrastructure and organization, respectively presented on Tables 2.2 and 2.3. On Table 2.2, features are focused on resources that allow flexibility to extend the tool, including its openness and extensibility by plug-ins, possibility to annotate (describe) basic components, share and reuse resources, schedule activities and provide some level of security. We use ‘+’ to indicate compliance, ‘$\pm$’ a partial or limited compliance, and ‘-’ the lack of it. On Table 2.3 we analyzed the following: the basic component (building block) of the system, how it can be associated to other components, and which resources can be used to define a component. The table also contains a description of flexibility to change the flow of execution and which contextual information is associated to components. Column “resources to define a component” gives an idea of a system’s flexibility to create/execute workflows on CIGs so let us now explain these systems.

Trident [9] is a scientific WfMS that provides workflow provenance, schedule, and monitoring. However, there is a lack of extensibility features, especially to deal with VisTrails [38] has resources for visualization and creation of workflows by analogy, useful to provide easy use for non-IT specialists. Also, there is an exemplification mechanism that allow faster identification of the purpose of a task, which benefits workflow creation to achieve the goal. There is a versioning tree that allows to view changes made to workflows, which would be interesting to analyze different interventions applied to patients. The tool has limitations regarding the use of subworkflows and support to share and retrieve workflows, reducing its suitability on collaborative environment.

Kepler [3] can execute tasks sequentially, paralleled, iteratively, etc. The WfMS supports a wide diversity of options to implement tasks, including webservices, R and XSTL. It is possible to register provenance information and to semantically annotate components, using URN (*Uniform Resource Name*), which is interesting to make links between semantically described clinical guidelines. The limitations found are the lack of a client-server architecture.

Taverna [40] does not have an intuitive interface, but is easy to extend and supports annotations, provenance and sub-workflows. Also, it allows a diverse use of resources to implement tasks, such as webservices, Java API and spreadsheets. Tasks can be organized hierarchically using sub-workflows.

ASBRU [67], part of Asgaard [85], allows the design and execution of tasks. The basic component of the work (a plan) can have different attributes and be composed by subplans, forming a hierarchy. Atomic units of plans are actions, which represent a specific tasks under a plan, and have the flexibility to be associated to a user interaction, external program or device. The work deals with context and provides flexibility to change and adapt to situations. It is done associating a set of attributes that are used to perform reasoning, trigger plans, change states and alter measure values. Such attributes are: preferences (constraints, e.g., strategy, utility and resources), intentions (goals), conditions (rules that govern state transition) and effects (known effects that plan arguments have over measurable parameters). From the infrastructure point of view of ASBRU and Asgaard do not provide a flexible way to extend using plug-ins or modular components from third party developers, and do not implement security policies.

[50] developed Careflow System, a WfMS developed for healthcare, using case handling technique to add the necessary flexibility inherent to the problem. The workflow can be executed in a flexible way because tasks are oriented to data and can be executed by different users. Each task has the flexibility to be executed, skipped or re-executed. As result of the case handling flexibility usage, the complexity to deal with context variables is transferred to users, who need to know about the case to deal appropriately with the task. To avoid the inadequate handling of tasks, there is an access control mechanism that associate users and tasks to roles. The system is developed over the YAWL WfMS, thus ensuring its important infrastructure features, such as reuse and share of workflows, flexibility to extend, client-service architecture, and flexibility to use external resources as tasks and support for subworkflows.

CPO [84] was developed focusing on healthcare applications. However, as the authors remark, their approach is different from those based on TNMs (e.g., ASBRU). "The tasks in clinical pathways are not decisions and actions recommended to clinicians, but the interventions to be performed by a multidisciplinary team by using healthcare resources, which contains not only clinicians but other healthcare professionals within one or more organizations." As result, CPO is more similar to a traditional WfMS, with tasks that can be composed by subtasks, as workflows and subworkflows.

Perikles [65], also extends YAWL to increase the flexibility of a traditional WfMS, and makes it more suitable for healthcare applications. To this purpose, the work adds resources to control and plan tasks. Each task is specified under the HL7 standard, and may specify which tasks must be executed before or after another one, guiding users under activity planing.

ClinicSpace [72] uses [83] middleware to provide a tool to support collaboration and management of resources. The work is developed for pervasive and context-sensitive computing, interacting with sensors, devices and users. One of the main features is the recommendation of tasks based on task execution log.

The research analyzed can be classified in four main groups: scientific WfMSs ([40],[38], [9], and [3]); business WfMSs ([65]); CIGs ([67], [84], and [72]); hybrid approaches that extend or use WfMS to create a healthcare application.

Analysed WfMSs are extensible and, mostly, comply with open standards to connect with services; they support the management of provenance and annotation and can handle long transactions. The main workflow standards and/or tools have big libraries of shared workflows and support routines. Additionally, they allow the use of abstract activities, which can be associated to different resources, such as tools, documents or algorithms – provided as source code or service. Moreover, they commonly provide a more complete execution infrastructure with support to client-server architectures and allow the reuse of already existing resources. However, they lack support for dynamic self-adaptation, and do not provide support to context changes.

CIGs are essentially activity graphs and may accept the use of external resources. CIG execution systems are focused on guiding healthcare professionals through recommended actions and register, in a database, data from executed tasks or events. They are usually hierarchical (e.g., TNM approach) and embed the practices and usages of health guidelines. Moreover, they support definition of sets of conditions which tailor the actions to perform for each situation. In fact, such sets are nothing more than contexts for CIG execution.

An important characteristic found on CIGs is the approach to support the methodology and the pattern of work of healthcare professionals. It is essential to reduce the resistance that professionals may have to the use of a new tool, as well as reduce the learning curve to use it.

There is, also, initiatives that combine CIG and WfMS: [22], [50], and [65]. Those tools are able to extend the properties of WfMSs, specializing its features to be used in healthcare domain. The result is the possibility to use the WfMS features we characterized as infrastructure, and the improvement of the tool to allow better usage for healthcare professionals. Some of the properties of WfMSs and CIGs, however, can be lost while processing a workflow or a guideline. [50], for example, add flexibility to workflows, but transfer to the user the responsibility to deal with context adaptations. [65] limits the use of external tools to those compliant with HL7 standards.

This comparative study guided our model of context, to extend scientific WfMSs. Figure 2.1 summarizes our perception as result of the comparative analysis presented in CIG systems. At the top, we show an abstract representation of the CIG approach to organize and handle tasks, as observed in CIG systems.

Each box represents a plan whose execution is determined by the fulfilment of conditions, evaluated according to the state of variables, which we classify here as context variables. By context we mean, for example, the patient condition or the outcomes of a procedure. As illustrated in the figure, each plan can be decomposed in sub-plans, which in turn also have conditions defined by context variables – contextual conditions. We contrast this approach to the workflow approach illustrated at the bottom of Figure 2.1. Workflows are organized as a flow of tasks, whose connections are depicted by arrows. Even tough workflows can decompose their tasks in sub-workflows, different from CIG systems, this composition is a reuse strategy and will not guide the workflow execution according contextual variables. This observation motivated our work of applying this CIG hierarchical decomposition, based on contextual variables, to workflows; resulting in our context-driven workflow mechanism, detailed in the next section.

This will be discussed further, using our real case study as a basis for nursing activities.

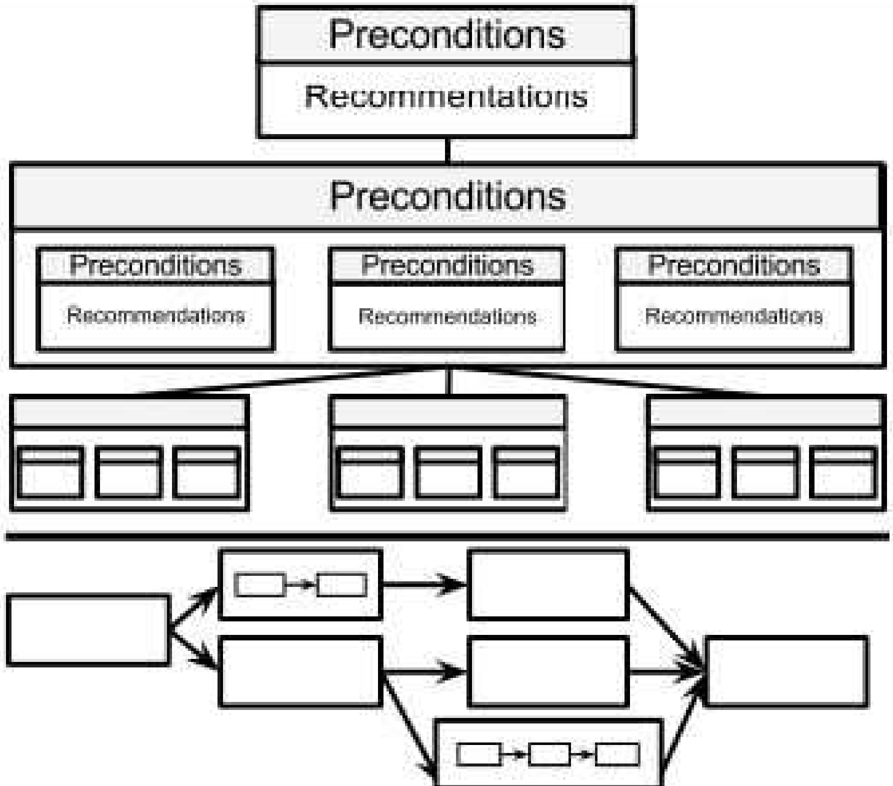

Figure 2.1: Organization of CIG and Workflow approaches as well as our proposed model.

2.4 Characterization of the scenario and proposal

Our case study involves the PROCENF system [58] and nursing professionals from the hospitals from University of Campinas and University of São Paulo [2]. Given this scenario, as well as the work described in [30], we identified that a patient's admission and monitoring process in a hospital can be expressed, in a general way, by the abstract workflow presented in Figure 2.2. This workflow reflects the patterns of the analysed CIGs, which are synthesized in table 3. The figure uses vocabulary from NANDA – North American Nursing Diagnosis Association – [41], NIC – Nursing interventions classification – [69], and NOC – Nursing outcomes classification – [19]. As can be seen, the workflow includes an iterative step in which a patient passes through anamnesis interrogation, an assessment phase. Then, there is the analysis of registered data to diagnose the problem and to identify expected outcomes (prognosis). Intervention planning and application phases occur to achieve an outcome. Outcome analysis consists in the analysis of intervention results, followed by updates to anamnesis records. If the treatment achieves

[2]The hospital complex of the University of Campinas alone receives about 500,000 appointments, with over 43,000 internments and 34,000 surgical interventions per year.

expected outcomes, the patient can be released. Otherwise, a new iteration occurs.

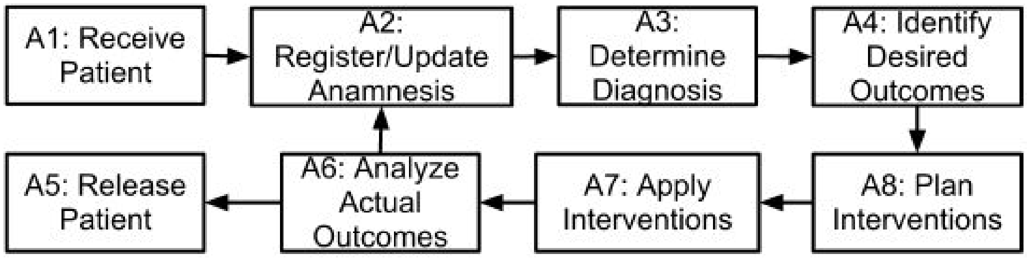

Figure 2.2: High level *Workflow* for healthcare treatment.

Health plans are characterized by progressively refinement by health professionals according to the context in which they are executed. While several "standard" procedures exist for a multitude of situations, each procedure is related to a given situation, or context, e.g., illness to be treated, patient anamnesis, and so on. However, depending on how a patient responds to interventions, a plan may change drastically. Thus, the execution of a sequence of tasks within a procedure is usually undertaken hierarchically: global procedures are defined in a high level manner, and undergo top-down refinement according to a given situation. Thus dynamics of this context-driven construction presents a marked contrast to other domains in which workflows or plans are conceived

If each procedure is defined as a workflow, this involves at least two aspects: (i) The execution of a given workflow may be suddenly interrupted to yield control to a different workflow dynamically defined by context and the original workflow may not even be ever resumed; (ii) Every workflow task subsumes subworkflows that are chosen according to context.

The dynamic change of tasks implies on another important aspect to be treated: traceability. Not only does traceability play a major role, in the sense that all action and actors must be recorded, but requires a new dimension – one must also keep track of dynamic configurations (how, when, and by whom). It is fundamental to provide a history of steps performed, to allow to learn with experience from other professionals and to recognize which steps were decisive to the achieved outcomes. Because of this, provenance is an important feature that must be supported.

The analysis presented in the previous section stresses the importance of guiding the workflow execution by means of contextual variables, as observed in CIG systems. Our proposed architecture is able extend a workflow engine to afford equivalent properties of a hierarchical decomposition based on contextual variables, as we will further detail.

Figure 2.3 shows the top level abstraction workflow to be executed, as portrayed in Figure 2.2. *CRec* is a dynamic data structure that records context variables at each instant. *CEng* is a context adaptation engine, an extension of a workflow engine, which monitors the context (*CRec*) and dynamically adapts workflow execution. The part shaded in gray represents the abstract activities yet to be executed under control of *CAEng*. The other abstract activities have already been refined and dashed line outlines the step executed under gradual refinement and execution.

Figure 2.3 illustrates two consecutive steps of a workflow execution carried by an engine, designed to support our context-based workflow customization and to allow to the user to interrupt or change the execution of an activity at any moment. In each step of the execution the *CAEng* monitors the running activity to capture context changes, updating the *CRec* (operation indicated by a dashed line). Whenever an activity finishes its execution, *CAEng* verify the new *CRec* state in order to apply modifications in the workflow (operation indicated by a gray area) if necessary. When the workflow starts, it has a starting *CRec* containing a set of basic context variables (see Figure 2.3 (i)), e.g., environment (e.g., type of medical facility), user, etc. The `Register/Update Anamnesis` is the main activity responsible for updating the *CRec*. Thus, the system can provide flexibility to change the flow of the execution, without force user to adapt the content to context.

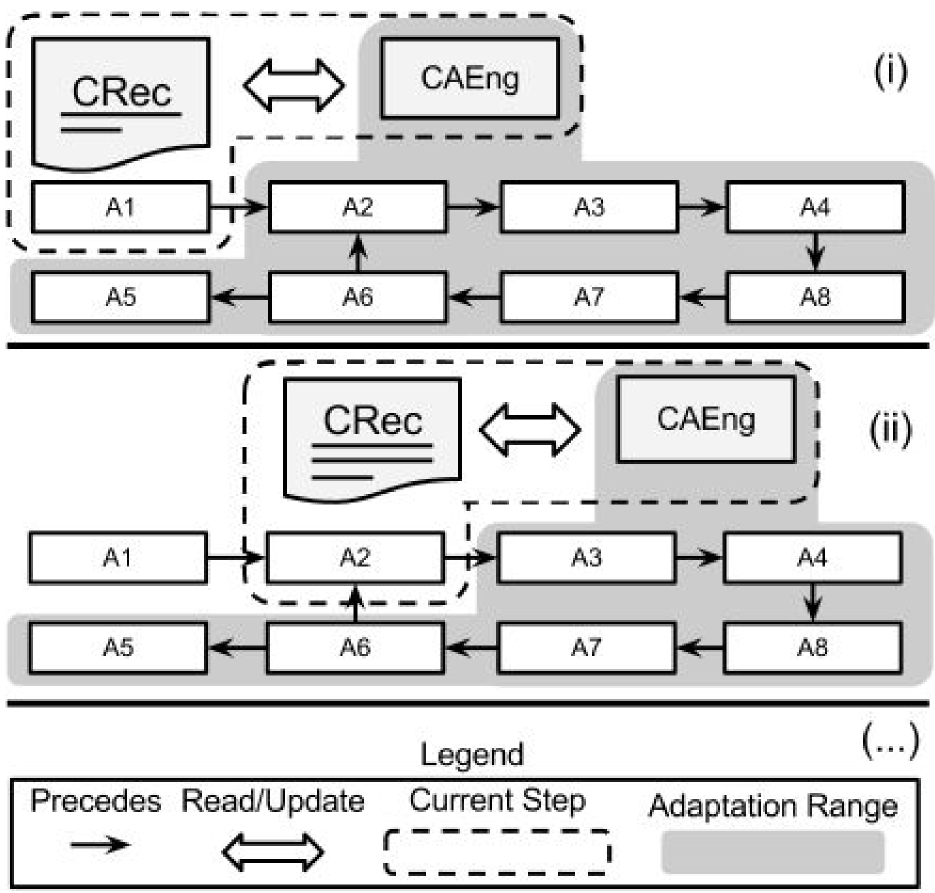

Figure 2.3: Adaptation of the flow of a workflow .

2.5 Conclusions

The addition of dynamic self-adaptation capability to scientific WfMSs can provide several advantages to healthcare activities. Benefits include automation and monitoring, and distributed execution of activities and the basis to support traceability of tasks and adaptations. Our work identified three important aspects that need to be considered: (i) Each task subsumes subworkflows that are conducted/invoked dynamically according to a context; (ii) Tasks can be interrupted or changed, adding flexibility to the way that health activities are performed under a WfMS; (iii) Traceability must be provided to record changes performed, as well as to allow the improvement of activities by of analysis the historic data.

Those three aspects allow: to extend a WfMS creating a tool that maintains the hierarchical nature of clinical guidelines, to adapt the flow to the context and to analyse whether performed tasks can be trusted or not. Also, the use of nursing standards contributes to better adaptation to healthcare workflows.

Future work includes the analysis and definition of context variables that will be used, resulting on a meta model and its instantiation. After this step, our model will be implemented in a scientific WfMS, which will be compared to the traditional approach, providing more evidences about advantages and disadvantages of the approaches.

Table 2.2: *Infrastructure characteristics of flow organized systems*

Work	**Open Source**	**Flexibility**	**Annotation**	**Client-Server**	**Schedule**	**Provenance**	**Reuse/Share**	**Security**
Trident [9]	+	-	-	+	+	Who executed, How long was execution, Associated data	Non integrated My-Experiment	User access and priviledge control and workflow roles
VisTrails [38]	+	+	+	+	-	Who executed; Where; How long; etc. Flexible framework to extend features.	DB share; Synchronization; Versioning made by external resources, e.g. SVN.	-
Taverna [40]	+	+	+	+	-	Open Provenance Model: Executed service;Execution date; Parameters used.	MyExperiment Integrated	Secure services and user credentials
Kepler [3]	+	+	+	-	-	Records provenance information	KAR files can be reused and shared. There is a central KAR repository.	-
ClinicSpace [72]	-	-	-	+	-	Records provenance information	Allows the reuse of registered tasks	Limits the user access to tasks
ASBRU [67]	-	-	+	-	+	Temporal View provides events history.	Import, export, and duplicate plans	-
CPO [84]	-	-	-	+	+	-	Model created as ontology to be reused	-
Perikles [65] extending YAWL	?	?	±	+	+	Logging mechanism from YAWL	Provides reuse	±
Careflow [50] extending YAWL	+	+	+	+	+	Logging mechanism from YAWL	Provides reuse	Limits user's actions by roles and authorization.
[22]	-	-	+	+	+	Logging mechanism from BPEL Server	Provides reuse	Protects processes and encrypts communication

Table 2.3: *Basic component (BC) characteristics of flow organized systems.*

Work	**Type**	**BC**	**Resources to Combine Components**	**Flow Flexibility**	**BC Organization**	**Modelling of Context**
Trident [9]	WfMS	Activity	Modified .NET classes; WebServices imported using external tool.	-	Sub-workflows	-
VisTrails [38]	WfMS	Activity	WebServices;Python Packages	-	Does not make clear that a workflow can be used as activity.	-
Taverna [40]	WfMS	Activity	WSDL/RESTful services, BioMart, BioMoby, SoapLab, Java APIs, R, Beanshell, Spreadsheets, Plugins to extend support.	-	Sub-workflows	-
Kepler [3]	WfMS	Activity	WSDL/RESTful services, R, MatLab, Spreadsheets, command-line applications, XPath and XSLT;	-	Sub-workflows	-
ClinicSpace [72]	CIG + middleware	Clinical Task	Limited to previously integrated tasks	Invoke tasks under certain conditions	Tasks and sub-tasks	Users, location, time and resources.
ASBRU [67]	CIG	Plans	User interaction and external tools and devices	Relationships and conditions change execution flow	Hierarchy	Conditions for plan activation
CPO [84]	CIG	Activity	Specification of interventions with associated values	?	Subprocess similar to subworkflows	?
Perikles [65] extending YAWL	CIG	Task	Service with HL7 compliant interfaces	?	Subworkflows	?
Careflow [50] extending YAWL	Case Handling + WfMS	Task	External applications, Java classes and webservices	Users can execute, redo, or skip task and change data at any time.	Tasks, Sub-Tasks and Workflows	Each task consider the entire case. Part of the context is handled by user
[22]	Ont. KB + BPEL server	Tasks	Web Service	Dynamic workflow composition and execution	Hierarchy	Yes

Chapter 3

Using Healthcare Planning Features to Drive Scientific Workflows on the Web

3.1 Introduction

The growth in the number of patients of a hospital brings the challenge of managing appointments, admissions and surgical interventions. There is the need to increase the efficiency and flexibility to manage associated data. The use of computational resources to address this challenge implies on more technical requirements. The possibility of using data available on the Web has added a new dimension to this problem, with additional heterogeneity factors.

Some researchers [76], in the context of chronic disease care, identified requirements that should be fulfilled by systems to be applied to the health domain, among which we single out: (i) Support for the shared needs and behaviors in care; (ii) Allow customization for disease-specific needs and to support the needs of different types of users; (iii) Explore new approaches for information input into the EHR (Electronic Health Record) as well as transfer, efficiently, data from medical devices into them.

Care-flow is at the core of healthcare management, involving directly or indirectly the requirements presented by [76]. As a consequence, one natural approach to start to deal with the problem has been to use Computer-Interpretable Guidelines (CIGs). Informally, a CIG is a specification in some kind of computer interpretable language that defines the flow of steps to be taken in each situation met by a health professional. Those guidelines manage the care-flow, customize needs and details for patient treatment and deal with data gathered, clinical guidelines, while preserves the rationale of healthcare professionals. Though CIGs are suitable for guiding the care-flow, from the perspective of the planning of the paths chosen, the tools that implement a CIG approach are still incipient, missing support to extensibility, reuse and share of content, collaboration among professionals and traceability.

Even though it is possible to define guidelines for “blocks of actions”, the whole process involved in the healthcare of a given patient is driven by the context, which is captured during the process itself. In a typical scenario, data from given care-flow step will define the next steps to be followed. The flow of actions *emerge* from the interaction between available blocks and the context. Therefore, one of the most successful approaches for CIG is the Task-Network Model (TNM) [57], which mimics this context-driven evolution. The process starts by a seed block of actions, which is unfolded according to context data collected during its execution.

Scientific workflows, on the other hand, are concerned with the planning and flow of tasks, but associated to scientific experiments and general tasks. They have been developed for years and tested on areas such as Astronomy, Biology, Gravitational Physics and Earthquake Science [13]. As a result, those tools have consolidated mechanisms to deal with large amounts of data, collaboration, extensibility of their resources, and association with external sources of data, tools and algorithms. In general, Scientific Workflow Management Systems (SWfMS) manage experiments' provenance, keeping detailed and meaningful records of data involved on experiments and processes that affect the data. Provenance is fundamental for scientific processes because it provides important documentation that is key to preserve the data, to determine data quality and authorship, and to reproduce as well as validate the results of such processes [23].

Workflow execution has been adopted where resources are distributed on the Web, helping the coordination and monitoring of processes. They are being increasingly used to create complex Web applications by Web service composition or by providing a thin client, accessible through a browser, to conduct large scale processes and experiments [81]. In order to enhance their applicability, research efforts are concerned with providing workflow mechanisms with more flexibility, including in the healthcare scenario such as [22], or [65]. Even though there are mechanisms that can be used to pre-define exceptions and alternative paths along workflows, they were not designed to dynamically evolve the workflow specification driven by the context, e.g., unfolding blocks of actions according to contextual data collected during a process execution.

Compared to CIG systems, workflow systems have been widely adopted and have consolidated standards and tools. In order to exploit the advantages of workflow systems in the health context, this work proposes to incorporate into workflows the dynamic and context-driven approach followed by care-flow systems. This is based on our experience [78] that shows that the effort to do this is less complex than the process to adapt CIG systems so that they can acquire the features that are causing scientific workflows to become widely adopted. This paper analyses the features that confer high flexibility to CIGs – mainly those based on the TNM due to its emergent behavior – and presents our initial approach to bring them to a SWfMS.

3.2 Clinical Practice Guidelines and Task-Network Models

Clinical Practical Guidelines (CPGs) are written guidelines that describe the evidence-based procedures to be followed during diagnosis, treatment, and clinical decision making for a specific disease. Their textual format can be easily diffused, but not easily used in daily work [56], because of a multitude of forms and specialized vocabulary. Moreover, they depend heavily on the expertise of health professionals.

An approach to solve these problems is the dissemination of guidelines' content in machine-interpretable representations, which are more suitable for use in individual clinical decision support systems [56]. This approach has led to the creation of Computer-Interpretable Guidelines (CIGs), which implement guidelines in active computer-based decision support systems. CIGs adopt models to represent the content to support decisions. Some examples of such models are Task-Network Models (TNMs), Medical Logic Modules (MLMs) and Augmented Decision Tables (ADTs).

[34] identified 8 knowledge models implemented, related to clinical decisions, concluding that TNM was the most commonly adopted. The authors characterize TNM in two ways: general and formal. General TNMs are "flowcharts or process maps without formal semantics". Formal TNMs are "guideline-based clinical tasks – actions, decisions, queries – that unfold over time, with a formal syntax and semantics". According to [57], TNMs succeed over alternative approaches, such as MLMs and ADTs, because they do not provide full support for conceptualizing a multi-step guideline that unfolds over time. Considering those aspects, in this work we focus on TNMs.

Figure 3.1 shows a usual structure of a TNM. As the 'Legend' (top right) shows, the basic component is a plan and it represents a procedure to be applied to some case. In each box, a rectangle with a label is the name of the plan. The top plan – labeled *Ventilation*– is the starting (seed) plan. The following plans (below) are sub-plans that can be triggered by the upper plan according to rules. To decide whether a plan should be applied or not, there are *Characterization Criteria* that define conditions to be analyzed in order to match the patient case to the diagnosis. Another aspect that can be used to determine if the plan is suitable to the case is the *Expected Outcomes*. The flow of actions and conditions appear whithin the *Procedures* box. Whenever a plan is recognized as a way to achieve the same result desired by the healthcare professional, the plan will be triggered.

A triggered plan will follow the *Procedures* that are recommended to be applied to the patient. The repetition of those internal procedures is guided according to the *Repetition* specification associated to the plan. Also, it is possible to associate *Abort Criteria* to interrupt the plan when a specific situation is achieved. A plan can trigger potential sub-plans, allowing to modularize and reuse plans that already encapsulate needed practices. The fact that the results can be unexpected, and consequently the sub-plans will be selected during the execution, pro-

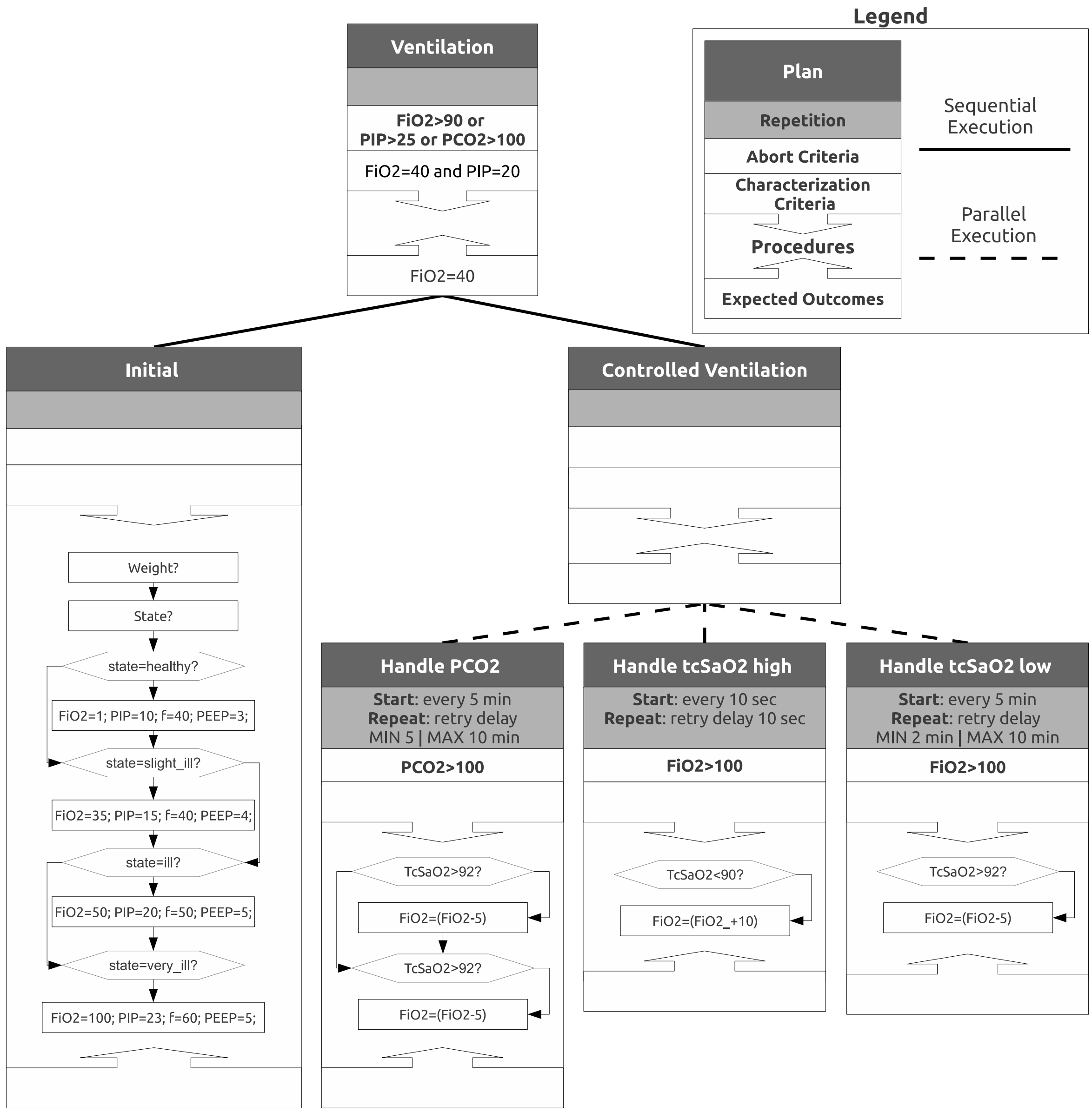

Figure 3.1: Example of CIG structure for a *Controlled Ventilation plan*.

duces an *emergent behavior* in the flow of activities. Several potential flows, constrained by rules, will dynamically shape one flow on-the-fly during the execution, interacting with contextual values. This work captures this rule-based and context-driven emergent behavior and adapts it to workflows.

To explain the characteristics of TNMs we use the same scenario that Aigner and Miksch [2] adopt to validate the CareVis platform: Infants' Respiratory Distress Syndrome (I-RDS). As Miksch [51] explains, "after I-RDS is diagnosed, a plan dealing with limited monitoring possibilities is activated, called *initial-phase*. Then follows, depending on the severity of the disease, three different kinds of plans, *controlled-ventilation*, *permissive-hypercapnia*, or *crisis-management*. Only one plan at a time can be activated, however the order of execution and the activation frequency of the three different plans are depending on the severity of the disease".

The scenario is shown on Figure 3.1. The plan execution order is read as top-down, left to right. Dashed lines from a plan represent alternative sub-plans that are used according to the conditions. To represent the internal `Procedures` , we use the same representation as CareVis: flow-charts. An internal procedure is described as a flow of CareVis 'single-steps'. Each single-step is either a variable assignment, a if-then-else construct (hexagon), an ask element (rectangle).

The *Ventilation* plan is activated when the characterization criteria *FiO2>40 and PIP=20* is satisfied. The plan and all sub-plans are aborted if the condition *FiO2>90 or PIP>25 or PCO2>100* is achieved. As *Ventilation* plan does not have internal procedures, the *Initial* plan is executed, following the sequential order specified. The *Initial* plan just has a set of internal procedures, such as ask *weight* and *state* of the patient and define variable values according to the state value (*healthy*, *slightly ill*, *ill* and *very ill*).

3.3 Workflows

A workflow is defined as the movement of tasks through a work process describing how tasks are structured, who performs them, the resources needed and their relative order [21]. On a computational context, a workflow specification can be seen as an abstraction that allows the structured composition of programs as a sequence of activities aiming a desired result [55].

Business and science are the two main driving forces behind the development of Workflow Management Systems (WfMSs). According to Sonntag et al. [71], business workflows are focused on the control of the flow, adopt agreed-upon communication standards, in order to facilitate interoperation between different software systems and companies, and commonly are concerned about fault handling, transactions, or quality of service features. Scientific workflows, on the other hand, are focused on data transformation, commonly may involve computation-intensive tasks, and are concerned with the specification of explicit data flow, the exact reproducibility of workflows, or processing of data streams.

The Workflow Management Coalition (WfMC) is one important participant of those driving forces behind WfMSs development. Because of its involvement on the earlier stage of the creation of workflows, it defines the workflow components under the business perspective. In this work we use the WfMC definitions, relaxing them to a broader perspective that also covers scientific workflows [82]. Here we differentiate between a workflow specification and its instantiation (an actual execution). Moreover, we use the term 'task' as a synonym for 'activity' (where an activity is "a description of a piece of work that forms one logical step of a process"

[82]. We likewise distinguish *Process definition* from *Process/Activity Instance*.

In order to help compare workflow and TNM approaches, we re-engineered the *Controlled Ventilation* plan of Section 3.2 using part of the graphical notation of flow-charts used on TNMs, but following a representation similar to that adopted by the Taverna [40] workflow system – see Figure 3.2.

The 'legend' (top right) shows the basic notion of structure of a process. The top rectangle with a text is the name of the process. The second one is the repetition criteria, which allows to define criteria to repeat the process and an interval for each repetition. Like TNMs, workflows may contain internal procedures, specified program code or instructions, that allows to perform activities. For instance, processes 'Handle PCO2' contains repetition criteria and embeds code that is repeated according to the criteria.

An important distinction between TNMs and workflows is the access to the variable values. While on TNM each value may act as a global variable, which can be accessed from anywhere in a plan, on workflows the values should be explicitly passed for each process through a port. Input ports receive the values of a process, while outport ports contains the result of a process. Of course, processes can read and write from a common storage, but this approach reduces the generality of processes.

To exemplify the use of a workflow, consider again the *Controlled Ventilation plan*, now executed as a workflow depicted in Figure 3.2. To determine whether the *Ventilation process* should be executed, the *Ventilation Activation Condition* process is introduced. If the condition *FiO2=40 and PIP=20* is satisfied, the result is passed to the *Pass Output port*, which indicates that the *Ventilation process* should be performed. If the condition is not satisfied, the result is passed to *Fail Output port*, finishing the execution of the workflow. Ports thus enable conditioning the path according to the obtained result.

Considering that the *Ventilation Activation Condition process* generated a *Pass Output* value, the *Ventilation process* receives an input value and passes it to its internal processes. As well as on Figure 3.1, *Initial* and *Controlled Ventilation* are sequentially executed. This execution is explicitly defined by the fact that the *Initial Output port* is connected to the *Controlled Ventilation Input port*. While the *Initial process* uses its internal script to obtain the values of *weight* and *state*, *Controlled Ventilation* has sub-processes that are performed on any order. In fact, differently from TNMs, those sub-processes can be executed at the same time. The split-join, represented by a circle before and after the processes, indicates that any process (among *Handle tcSaO2 low*, *Handle PCO2* and *Handle tcSaO2 high*) can be executed. The split-join allows to execute different processes and then combines the results so that can be passed to another process or port.

If we proceed mapping a TNM specification to a workflow, the alternatives in each stage will grow exponentially. This effect shows the limits of trying to directly map a TNM specification in a classic workflow specification. Our proposal addresses this problem. It blends the TNM rule-

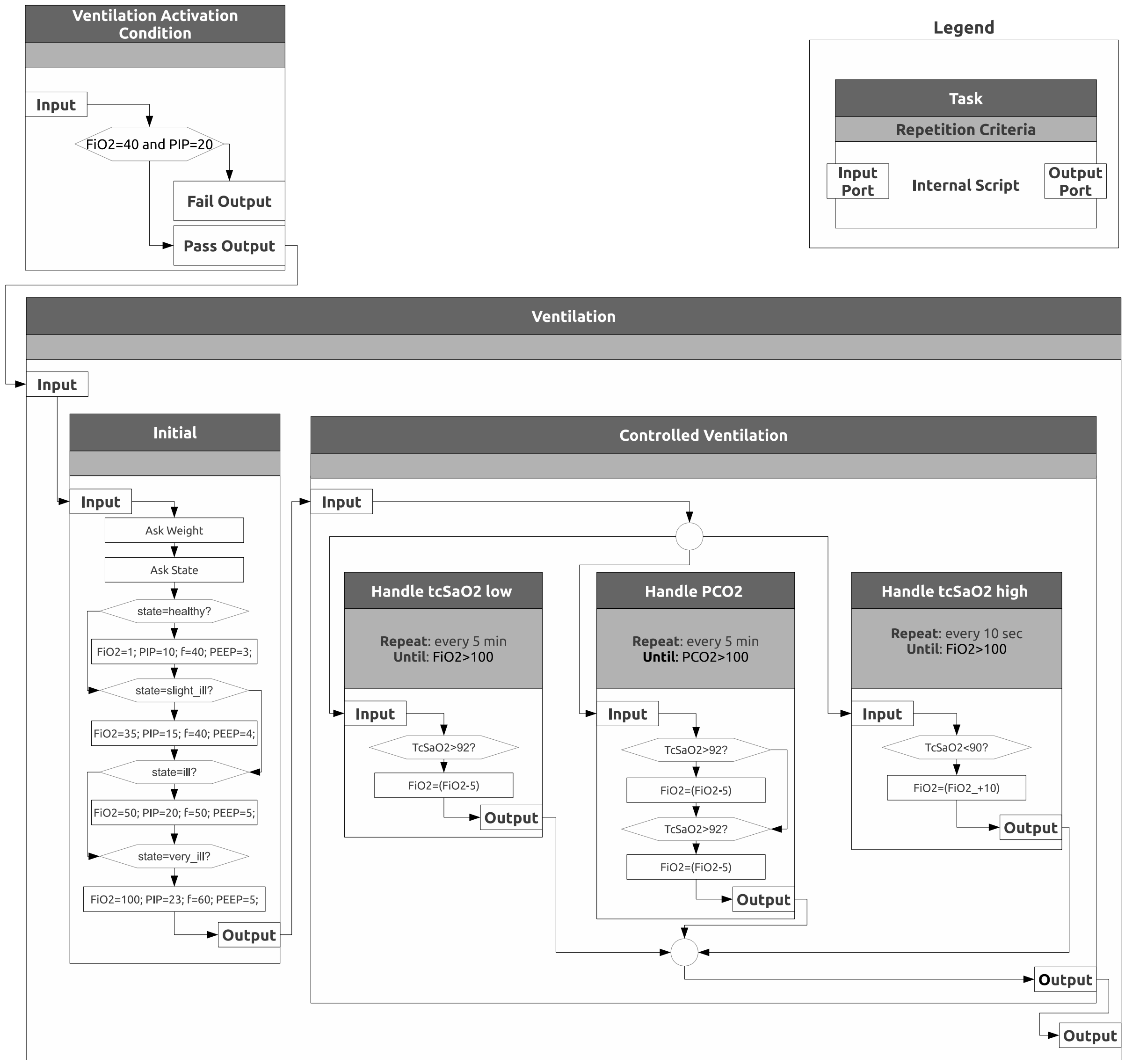

Figure 3.2: *Controlled Ventilation plan* re-engineered by us as a workflow structure.

based context-driven mechanism in a workflow system, providing an equivalent TNM ability of adapting itself according to the context.

3.4 Data-driven vs Process-driven approaches

In order to summarize some of the main distinctions between the Workflow and TNM approaches is: scientific workflows are process driven and TNMs are data driven. Thus, bringing TNM characteristics into scientific workflows will make the latter both data and process driven.

TNMs are built to reduce the complexity to deal with large amounts of content that must be interpreted and to guide the use of procedures that are recommended to achieve some state. Because a patient may change his/her condition unexpectedly, TNMs allow to activate or deactivate content (guides and procedures) to treat the current state of the patient. All guides and procedures are pre-specified according to the recommendations of medical councils and well established procedures.

Scientific workflows are focused on processing high amounts of data. Usually there are two common situations: (i) a scientist wants to represent and automate a well known/consolidated experiment and execute it with different data (parameters) with none or few changes during the execution; (ii) the scientist wants to create an experiment to test a new hypothesis, so (s)he starts to build a workflow including and changing processes and parameters after different executions to analyze the results.

The difference between both approaches can be observable on the basic component of each one. TNMs have plans that associate content, procedures and other plans. Also, there are criteria that must be satisfied to activate the plan; such criteria allow selecting a plan over all other possible situations that may occur. Scientific workflows have processes that can be associated to others to create the flow of data and transform/process it. The data that is passed from one process to another can lead to different paths, but commonly the possible number of paths and changes are not as high as on TNMs. When a new situation occurs and there is a need to incorporate a new process, the scientist changes the workflow – e.g., choosing from a repository of processes, including an external tool or creating a new process.

Another factor is that, to deal with all resources involved, it is necessary to provide tools to visualize and filter information, as well as be aware of the origin and the involvement of resource with respect to the data. A key issue is to maintain provenance information on data and procedures – e.g., what kind of data was used where, how and by whom. Here, workflow systems already provide some sort of traceability mechanism.

In this paper we focus on the main features required to achieve a data-driven workflow: changing workflow tasks to adapt them to a situation. Our approach is presented next.

3.5 Towards making scientific workflows data-driven

As presented in previous sections, a key factor to produce workflows able to adapt to the high dynamic health environments is the rule-based TNM approach, which triggers modules according to context values, producing an emergent behavior. Different from TNM specifications, which start from a seed and expands the flow, a workflow has a predefined starting flow. Therefore, we translated the mechanism of *unfolding* new activities on-the-fly according rules (TNM) to a mechanism of *adapting* the existing workflow on-the-fly according to rules (our approach).

In order to fully support dynamic workflow changes according to context, and give more flexibility to workflow execution, we decided to adopt reasoning capabilities and combine them with domain semantics. The solution found was to (a) use ontologies as the basis for workflow and concept representation, and (b) create adaptation rules to represent experts' knowlege. An *adaptation rule* characterizes a situation (cause) and defines the change (consequence) that should be performed to workflow so that it can be adapted to the context.

Given that there exist distinct workflow representations and formats (according to the WfMS adopted), the overall dynamic adaptation cycle can be described as follows. The workflow execution on the Web is monitored at each activity.Every new workflow state is mapped to an ontology instance, to which reasoning is applied, resulting in a new ontology instance (containing recommended modifications to that state). This new instance is transformed back to the workflow representation of the WfMS adopted, which is shown back to the user on the Web interface, to be validated or modified.

The architecture of our work is presented in Figure 4.1, where CRec (Context Record) is the linearized representation of an ontology, representing a workflow state. The main components are the following. A **Scientific Workflow Management System** (SWfMS) used to create and execute workflows, through a Web Interface, extended to incorporate the adaptation mechanism. A **Context-Aware Extension**, responsible for monitoring workflow execution and identifying the events that occur during that execution. A **Semantic Mapper** that is responsible for translating a workflow to a CRec and vice-versa, allowing the *Context-Aware Extension* to update the original workflow on SWfMS. An **Ontology** that specifies the main classes used to represent workflow activities and their parameters. Also, there are additional classes used to complement activity information (such as activation status) and associated content (for instance, patient data). A CRec (resp. CRec') is a **Context Record** that is an ontology instance that represents the current state of a workflow and all information regarding the patient under care. A set of **Adaptation Rules** (*AR*), written in SWRL, that will be used with an Ontology and a CRec to identify the changes recommended to the current situation. A **Context Adaptation Engine** (*CAEng*) that uses an Ontology Reasoner to process the *AR* combined to an Ontology and a *CRec*. The reasoning updates *CRec* to guide the adaptations that should be made on workflow.

The architecture enables dynamic adaptation of workflows to a context as follows. A SWfMS is accessed through a Web Interface (Figure 4.1, interaction 1). A workflow is assigned to be executed on the SWfMS (on interaction 2) and the SWfMS loads it (interaction 3) and updates the Web Interface (interaction 4). A user requests the SWfMS to start the executing workflow (interactions 5 and 6). When an activity is performed, the *Context-Aware Extension* captures the workflow state (interaction 7) and passes it to the *Semantic Mapper* (interaction 8). Next, the *Semantic Mapper* uses the workflow state to create an ontology instance that represents it (interaction 9) – *CRec*. The *CAEng* combines *AR*, *Ontology*, and *CRec* to identify the adaptations recommended to the workflow and apply them to *CRec*, creating a new version of it (*CRec'*. The *Semantic Mapper* receives *CRec'*(interaction 10) and maps its data to the workflow

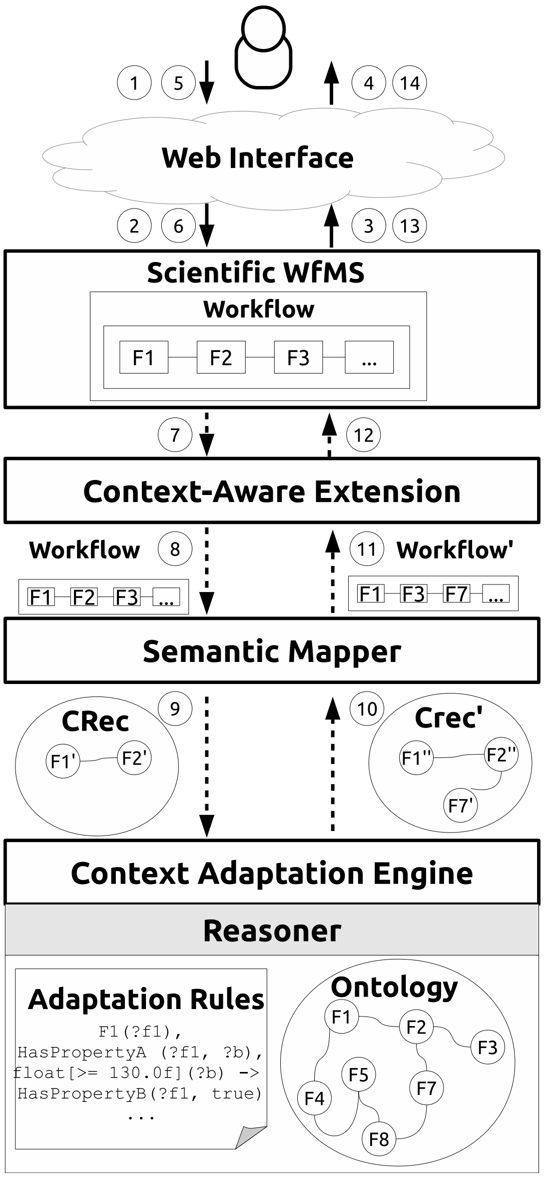

Figure 3.3: Overview of context-based workflow adaptation.

representation, passing it back to the *Context-Aware Extension* (interaction 11). The workflow on the SWfMS is updated by the *Context-Aware Extension* according to the new workflow (interaction 12). The SWfMS updates the Web Interface (interaction 13). Finally, the user sees the adapted version of the workflow (interaction 14).

Our architecture may be used on different scenarios and on different SWfMS, but its components created must be changed according to the scenario (e.g., rules and ontology customization) and platform adopted (e.g., Semantic Mapper).

3.6 Instantiation on the Web

We applied the principles behind our architecture to the context of nursing care to attest the flexibility of execution of the workflows. The tasks used to construct the ontology and the rules were based on the PROCEnf system [58], developed and adopted by the hospital of the University São Paulo. PROCEnf is also in process of adoption at the our University Hospital[1]. Like most such systems, PROCEnf is heavily centered on offering health professionals sets of forms to fill.

We chose PROCEnf among other reasons, because we can reuse its components. Moreover, given our need to validate our implementation with actual users, this choice offers to the nursing staff of our University Hospital a set of forms and vocabulary they are familiar with.

Based on PROCEnf and on the work described in [30], we described the flow of a patient's admission and monitoring process in a hospital. In a general way, the process includes an iterative step which is started by an anamnesis interrogation, which is an assessment phase to evaluate the patients' conditions. Next, there is the analysis of recorded data to diagnose the problem and to identify expected outcomes (prognosis). Based on those expected outcomes, health interventions are planned and applied. To identify whether the outcomes were achieved or not, the intervention results are analyzed and the amamnsesis records are updated. If the treatment achieves the expected outcomes, the patient can be released. Otherwise, a new iteration occurs.

To model PROCEnf forms as workflow activities, we created an ontology. In our current stage of research, we have about 30 activity classes involved with the diagnosis phase. Those classes are directly related to the PROCEnf system. As can be seen on classes 'CardiacFunction' and 'VitalSigns', there are different attributes that can be filled to draw a diagnosis. Depending on how an attribute is filled, distinct decisions have to be taken – i.e., the workflow has to be changed dynamically.

Another issue we faced was the integration of workflow structures, rules, and reasoning capabilities. As explained before, this was solved by treating all information within a single ontology-based reasoning framework. The *Semantic Mapper* transforms the workflow structures (together with a current workflow state) into an ontology instance, that is enhanced with the current patient state. This is then treated by the reasoner as any other ontology, using adaptation rules that encode domain and user knowledge. Such adaptations are performed based on the adaptation rules we created in SWRL. Those rules can be classified into four types: I) **Propagate field (parameter) value**: repeat the value of a field to related fields, avoiding the need to fill the value in other forms. II) **Infer field (parameter) value**: the value of a field according to the value of other fields. III) **Change form (activity) position**: the order in forms

[1]The hospital complex of the University of Campinas alone receives about 500,000 appointments, with over 43,000 admissions and 34,000 surgical interventions per year.

Table 3.1: Adaptation Rules

Rule	Antecedent	Consequent	Type
R1	CardiacFunction(?cf), VitalSigns(?vs), systolicBloodPressure(?vs, ?x)	systolicBloodPressure(?cf, ?x)	Propagate Value
R2	VitalSigns(?vs), systolicBloodPressure(?vs, ?x), float[>= 130.0f](?x)	hasSystolicPressureOutOfExpectedRange(?vs, true)	Infer Value
R3	VitalSigns(?vs), isActive(?vs, true), hasSystolicPressureOutOfExpectedRange(?vs, false), CardiacFunction(?cf)	position(?cf, 99)	Decrease Priority
R4	VitalSigns(?vs), hasSystolicPressureOutOfExpectedRange(?vs, true), isActive(?vs, true), CardiacFunction(?cf), position(?vs, ?po)	position(?cf, ?po)	Increase Priority
R5	ValuesAndBeliefs(?va), Evaluation(?e), hasPatient(?e, ?p), isActive(?e, true), hasAge(?p, ?age), integer[<= 12](?age)	isRemoved(?va, true)	Remove
R6	ValuesAndBeliefs(?va), Evaluation(?e), hasPatient(?e, ?p), isActive(?e, true), hasAge(?p, ?age), integer[>12](?age), isRemoved(?va, true)	isRemoved(?va, false)	Include

which are filled can be changed, increasing or decreasing filling priority, making related forms closer and unrelated or unlikely forms more distant in the filling order. IV) **Include/Remove form (activity)**: a form can be removed when a field makes it incompatible with the purpose of the form (e.g., pregnancy issues related to men). Inclusion occurs whenever a situation makes a previoulsy removed form viable.

Table 4.1 presents a subset of the rules, highlighting the types and some of their potential. The rules are specified in SWRL and are specified in cause/consequence form. The cause (antecedent) can be defined according to the value of fields and forms, by the composition of different predicates. A consequence (consequent) is the result of checking the antecedent, e.g., changing of a property value, including or removing values. For instance, rule *R2* says that when the *Systolic Blood Pressure* value of *Vital Signs* is large than 130, then the field *Has Systolic Blood Pressure Out of Expected Range* should be true.

Let us exemplify how adaptation rules are used in the reasoning process. Consider the following scenario: a nurse should fill a set of forms about different patient conditions, including vital signs, self-care, and comfort. At the beginning of the process, the professional fills a form about *Vital Signs*. If the value of *systolic blood pressure* is filled with value large than 130, rule *R2* (Table 4.1) has its antecedent condition satisfied, so the inference engine applies the consequent which characterizes the patient as having a *systolic pressure out of expected range*. As a result, two other rules are to be executed, leading to the new adaptations: *R1*, to propagate the value of *systolic blood pressure* to the *Cardiac Function* task and *R4* to increase the priority of the *Cardiac Function* task, which will put this form closer to the *Vital Signs* form. This order change highlights the need to provide additional information about Cardiac Function details and to avoid the loss of focus and information regarding this problem.

3.7 Conclusions

In this paper, we presented a proposal to adapt scientific workflows to the context of healthcare management, and its Web implementation. Our modifications to an SWfMS place such systems closer to the domain of healthcare, thanks to the inspiration from CIGs and to the fact that such systems, thanks to the data-driven flow, are suitable to dynamic scenarios.

We designed the model in which an SWfMS is extended by a layer which uses ontologies combined to rules as a means to make scientific worflows more data-driven than just process-driven. This in turn, brings more flexibility to the execution of the tasks as well as allows to choose whether a task should be executed according to more sophisticated conditions.

Future steps of this research will cope with TNM features, presented in Section 3.4, working on the better graphical integration and adaptation of the work with the SWfMS, allowing to: (i) Integrate the ontology content to the workflow tasks, allowing users to navigate through the concepts of ontologies and use those concepts also as a way to filter tasks and conveniently find resources on SWfMS; (ii) Provide support to the hierarchical organization of workflow tasks, giving more dynamic flow to task execution as well as making it more similar to the TNM approach.

Chapter 4

DynFlow - a software architecture for dynamic adaptation of workflows for healthcare planning

4.1 Introduction

Automated healthcare planning systems are designed to support the dynamicity of health environments, in which changes occur constantly as a patient's treatment progresses. This dynamic adaptation mechanism is based on managing blocks of activities, triggered and combined according to contextual data, producing a plan, which emerges from the interaction between these blocks and the context. A *context*, in this paper, concerns the patient under treatment and is defined by a set of variables that describe that patient – e.g. clinical data, history data and so on.

One of the most prominent approaches to model care-flow is the Task-network Model (TNM), which "decomposes guidelines into networks of tasks unfolding over time" [84]. It can be seen as a hierarchical directed graph that specifies a flow of activities. Its enactment is often supported by artificial intelligence planning environments, in which a TNM is specified using some adaptation of a goal-based planning language.

In spite of their modeling and flexibility capabilities, tools that implement care-flow systems are still incipient, missing support for features like reuse of content, collaboration among professionals and traceability of treatment procedures. On the other hand, these features can be found in workflow systems, widely used in a variety of environments, with consolidated standards and technologies. However, workflow systems are not well suited to address the dynamicity of healthcare environments, which are context-sensitive.

"A critical challenge for any Workflow Management System (WfMS), in a real clinical setting, is its capability to respond effectively when exceptions occur [56]." WfMSs must moreover meet other requirements to allow their use by healthcare professionals. They must support the rationale behind the procedures in the health context, allow the use of technical terms from the healthcare domain, use those concepts to guide care-flow and better respond to new information about patients.

There are initiatives that have investigated running workflows to automate healthcare tasks, e.g., [22], [50], or [65]. However, none of these initiatives considers dynamically adapting workflows to a context based on TNM features, which already incorporates the rationale of healthcare professionals and are developed to met the dynamics of healthcare area. Our work is focused towards filling this gap.

This paper presents DynFlow – our software architecture that extends workflow systems with the emergent and context-driven approach of care-flow systems, through blending TNM characteristics. Besides bringing to workflows a "way of thinking" of health professionals, the TNM is the key to enable context-aware adaptation properties in workflows. This blended model and architecture are the main contributions of this work. As far as we know, no previous work exploited such synergy between the workflow and TNM models. We have implemented DynFlow for the healthcare domain, as a Web application, thereby simplifying its use by healthcare professionals, who do not need to install any specific software to use it.

4.2 DynFlow – Architecture for Adapting Scientific Workflows

It is a consensus that automated systems are important to document, manage and control the flow of tasks in a health environment. Even though workflows have a long story of success in automating flows in several domains, healthcare offers additional challenges. Holland [36] states that complex adaptive systems cannot be modeled by a single governing equation or rule; rather, they are modeled by several smaller interacting parts governed by their own rules. This was confirmed by a study of ours on TNM-based healthcare systems [78].

Healthcare systems have not yet reached the maturity found in workflow systems, which have proved to be a successful solution in several business and scientific contexts. There is a big community around workflows conducting long term research involving, for example, traceability, provenance, reuse, Web-enabled representation and distributed execution. This work shows that we can enhance workflows to meet healthcare requirements, by blending their model with TNM properties.

Our response to the challenge of dealing with distinct (internal) workflow representations using TNM-like hierarchical structures, and adapting such workflows on-the-fly, is based on two complementary concepts: (a) adopting an ontology to define the basic representation structure used in DynFlow, and (b) defining rules to support reasoning for dynamic adaptation. On the representation side, the use of ontologies allows encoding expert vocabulary and knowledge into a workflow specification; we map workflow concepts into an ontology, which is extended with healthcare concepts, and instantiate each step of a workflow execution as an instance of that ontology. *Adaptation rules* encode an expert's domain knowledge. An adaptation rule, similar to production rules (e.g., in artificial intelligence systems), is stated as a pair (cause, consequence). It characterizes a situation (cause) and defines the changes (consequence) that should be applied to a workflow to adapt it to the current context. Rules can determine, for instance, relative task ordering, whether a parameter should be disseminated across other tasks, or generation of new values to be used in subsequent steps.

Figure 4.1 presents the overview of DynFlow. Its main components are:

1. Scientific Workflow Management System (SWfMS) responsible for executing care flow tasks, under guidance of healthcare professionals.

2. The Adaptation mechanism, the core of the context-sensitive adaptation process, composed of a Semantic Mapper and a Context Adaptation Engine.

3. A Monitor, responsible for identifying events that occur during workflow execution (e.g., start and end of activity execution) and for mediating the communication between the SWfMS and the Adaptation Mechanism.

In more detail, the Adaptation mechanism: receives information about events and context from the Monitor and changes the workflow according to the adaptation rules available. It is divided in two modules:

- Context Adaptation Engine – applies reasoning, using adaptation rules, to dynamically adapt workflows to a context. The entire process is based on ontology management and reasoning on ontologies. This representation choice helps context adaptation and provides a single means of representing workflows, rules and contexts, and processing them all together. This engine performs forward-based reasoning, as opposed to backward-based reasoning [4, 17].

- Semantic Mapper – Transforms information provided by the Monitor into a workflow representation recognized by the Adaptation Engine (and vice versa). This representation corresponds to Context Records (CRec) – see below.

The Context Adaptation Engine has four main components:

- An *Ontology* that specifies the elements to represent workflow components (activities and parameters). It is extended with domain information that helps process contextual elements. During workflow execution, the ontology is progressively adapted to the execution states, via inferences and instantiation of classes and properties.

- A *CRec* (*Context Record*) – a data structure that linearizes a workflow state[1] and all information regarding the case being run. The linearization is materialized into an instance of the Ontology.

- A set of *Adaptation Rules* (*AR*) that are used with the Ontology and a *CRec* to identify the changes recommended to the current situation. Associating *Ontology* and *CRec* to *AR* corresponds to combining the description of the domain, characterization of the current situation and the rules that guide what should be done.

- An *Ontology Reasoner* that is responsible for generating inferences from *AR* over *Ontology* and *CRec*.

DynFlow enables dynamic adaptation of workflows to a context as follows:

- A workflow is assigned to be executed on the SWfMS.

- A user requests the SWfMS to start executing the workflow (Figure 4.1 on interaction 1).

- When an activity is performed, the *Monitor* captures the workflow state (interaction 2) and passes it to *Semantic Mapper* (interaction 3). Next, the *Semantic Mapper* transforms the workflow state (interaction 4) into its internal representation *CRec*.

- The *Reasoner* combines *AR*, *Ontology*, and *CRec* to identify the adaptations recommended to the workflow and applies them to *CRec*, creating a new version of it (*CRec'*). This contains the state that the workflow should have to be adapted to the current situation.

- The *Semantic Mapper* receives *CRec'*(interaction 5) and maps its back to the new workflow state, Workflow', passing it to the *Monitor* (interaction 6), which forwards Workflow' (interaction 7) to the SWfMS.

- Finally, the user sees the adapted version of the workflow (interaction 8). At this moment, the user can agree to the adaptation or not. In the latter case, the original workflow continues its execution until the next activity is activated (going through another cycle of interactions).

4.3 Instantiating DynFlow

To validate the ideas behind the architecture, we instantiated it as a Web application. The code was written in Java 2 Enterprise Edition, using Java and JSP. The front-end was built using HTML, CSS and JavaScript. The main APIs used were Jena, OWL API and Pellet Reasoner. The *Ontology* was created using Protégé.

[1]The workflow representation includes activities, data instantiation, transitions, parameters and so forth.

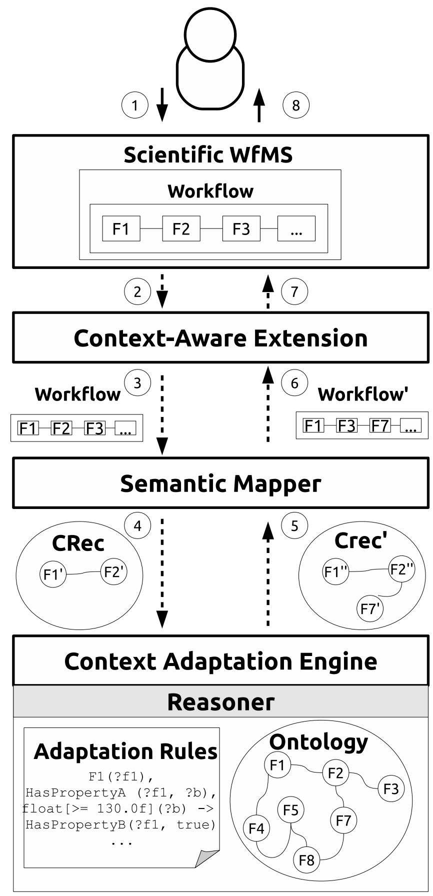

Figure 4.1: Overview of DynFlow for context-based workflow adaptation.

4.3.1 Case Study

Our case study was designed with help of nursing professionals and faculty from the University of Campinas Hospital[2]. Workflow processes were designed based on the PROCEnf system [58], developed and adopted by the hospital of the University of São Paulo, and in process of adoption at the University of Campinas Hospital.

PROCEnf is used to document nursing care procedures, for professional nurses and nursing students. It follows standards NANDA – North American Nursing Diagnosis Association – [41], NIC – Nursing interventions classification – [69], and NOC – Nursing outcomes classification – [19].

[2]The hospital complex of the University of Campinas receives about 500,000 appointments yearly, with over 43,000 admissions and 34,000 surgical interventions per year.

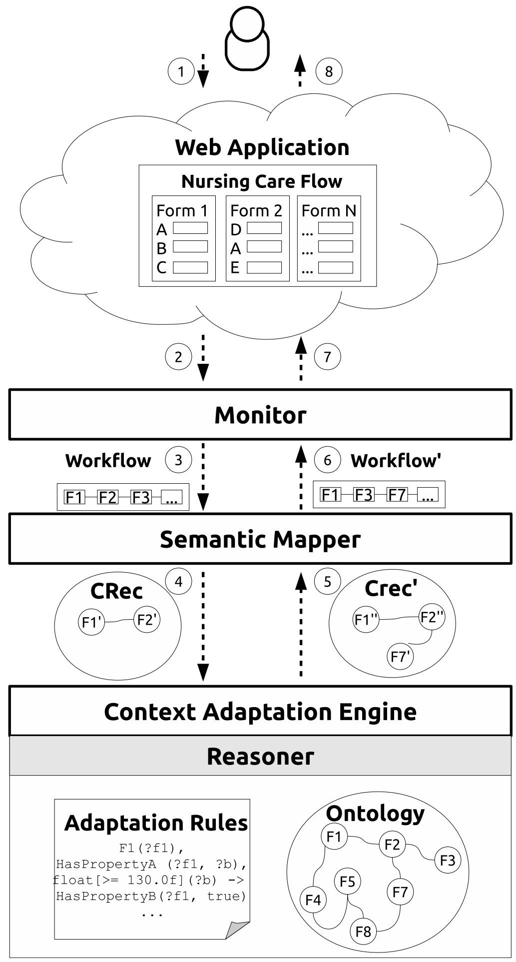

Figure 4.2: Creation of a lightweight workflow as a Web application using DynFlow for context-based workflow adaptation.

4.3.2 Implementing the Case Study

In this paper we focus on the implementation of the dynamic, on-the-fly adaptation of the anamnesis interrogation (as part of the nursing process). Figure 4.2 illustrates the instantiation of DynFlow. From the user's viewpoint, workflow activities and parameters are seen as forms and fields to be filled in a nursing care process. The *Web Application* replaces the SWfMS, and can be seen as a lightweight version of a SWfMS.

To express form processing within a workflow, we modelled PROCEnf forms as workflow activities. As mentioned in Section 4.2, we decided to model all domain concepts and workflows as ontologies. We adopted the OWL-S Semantic Markup for Web Services [48] to represent our workflow.

The Ontology has about 30 processes involved with the anamnesis phase, which are instances of the OWL-S AtomicProcess class. This kind of process corresponds to an atomic task

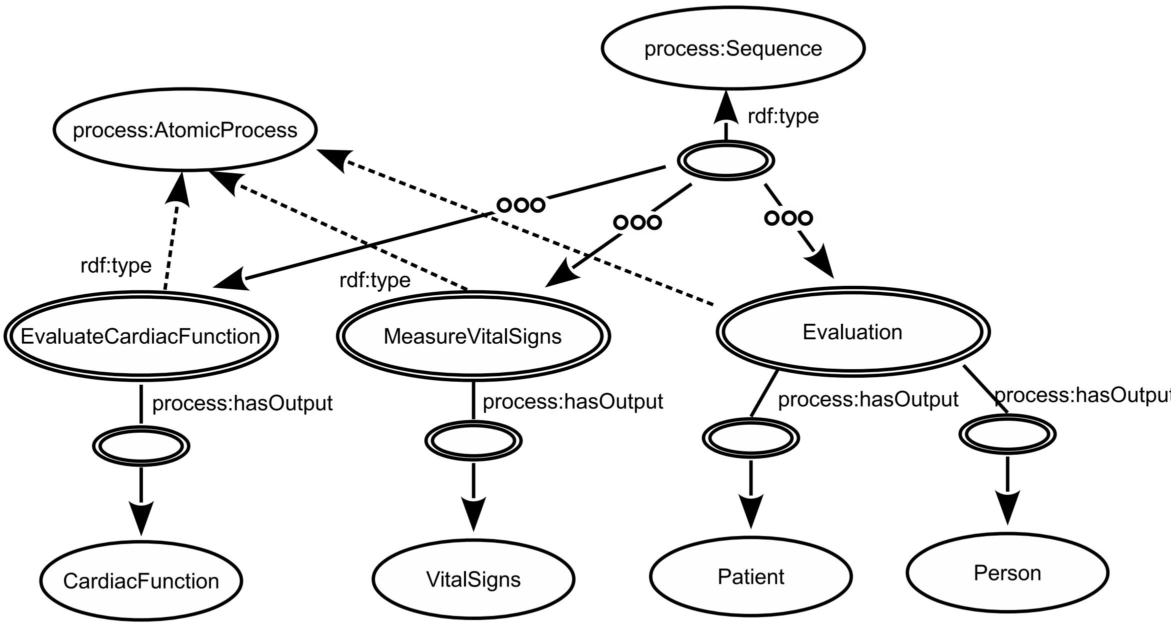

Figure 4.3: Representation of an ontology related to Nursing Care.

executed in the workflow, i.e. it will be subdivided in subtasks in the workflow perspective. Figure 4.3 shows a segment of an OWL-S PROCEnf workflow modeled as a sequence of activities (members of a process:Sequence object).

Classes such as 'EvaluateCardiacFunction' and 'MeasureVitalSigns' correspond to procedures which present forms to be filled during anamnesis. Each atomic process produces a data object – indicated by the process:hasOutput property – containing the fields of the forms. Since the combination of fields and forms is high, it is possible that some data or relations will not be noticed by the nurse, leading to mistakes. Dynamic workflow adaptation represents the context of the patient, helping such cases – e.g., by changing form order to prioritize activities.

Adaptations are performed based on the adaptation rules which we created in SWRL (Semantic Web Rule Language) and affect data and the OWL-S workflow. We implemented rules at two levels:

- **Parameter adaptation**:
  - Propagate the parameter's value to the subsequent activities and data objects.
  - Infer the parameter's value considering other values and activities.
- **Activity adaptation**:
  - Dynamically changes an activity's order in a workflow to better suit the context.
  - Activate or deactivate activities according to context

The rules implemented aid nurses by avoiding the rework of repeating data already filled, deriving data that can be inferred, making relevant forms more noticeable and irrelevant forms less noticeable. Table 4.1 presents a subset of the rules implemented, highlighting their types. Rules are specified in SWRL in cause/consequence form. The cause (antecedent) is a predicate defined on the value of data objects, and can be composed of several clauses. A consequence (consequent) is the result of checking the antecedent. For instance, rule *R2* computes the BMI *Body Mass Index* of a patient, given height and weight entered by the user, in the VitalSigns field of a data object.

The main functional requirement of our application is "Form Filling". Each workflow activity corresponds to a request to fill a form. On the top right of Figure 4.4, there is a box which shows a list of items (activities) in the suggested order of forms that can be filled. Each item is a workflow activity. The label at the top, 'Vital Signs', is the name of the data object produced by the activity, and its fields ('Weight', 'Height', etc.). At the bottom of the screen, buttons Next' and 'Clear' allow, respectively, to save the data and clear all field values. Users can select a form to fill at any time, clicking on that list.

If the 'Next' button is pressed, the *Monitor* (Figure 4.1) identifies the interaction and sends the workflow state to the *Semantic Mapper*, which transforms it into an individual of the ontology class (*CRec*), d the form fields become property values of that individual. Then, the *Context Adaptation Engine* combines *Ontology* and *Adaptation Rules* to generate a new version of *CRec* (*CRec'*). That version contains replicated parameter values, inferred parameters, activities activated/deactivated and the new activity order suggested.

The Semantic Mapper then transforms *CRec'* back into a new workflow state, which is forwarded to the *Monitor*. The *Web Application* page is reloaded, filling the form field values and changing the suggested order of activities.

Let us now exemplify adaptation rules. Consider the workflow on top of Figure 4.6. The nurse has already filled fields stating that the patient has age 16, weight 1.78m , height 51kg and is male. When the nurse presses the 'Next' button, several dynamic adaptations occur, such as:

- The BMI (Body Mass Index) is computed and entered automatically (rule R2).
- The BMI value is propagated to form 'Nutrition' (Rule 11).
- The BMI value is considered low for a 16 year old male. As a consequence, the order of activities 'Development and Growth' and 'Nutrition' is changed (rules R9 and R10).

The workflow generated as result of this adaptation appears at the bottom of Figure 4.6. Here, the BMI field value of activity Nutrition is already set and the order of activities of this workflow is changed.

Figure 4.4 is a screen copy of our prototype presenting the *Vital Signs* form before the adaptation, while Figure 4.5 shows the form after adaptation. This second figure illustrates two

Table 4.1: Adaptation Rules for nursing care diagnosis process

Rule	Antecedent	Consequent	Type
R1	CardiacFunction(?cf), VitalSigns(?vs), systolicBloodPressure(?vs, ?x)	systolicBloodPressure(?cf, ?x)	Propagate Field Value
R2	VitalSigns(?vs), height(?vs, ?h), weight(?vs, ?w), multiply(?hSq, ?h, ?h), divide(?bmi, ?w, ?hSq)	hasBMI(?vs, ?bmi)	Infer Field Value
R3	VitalSigns(?vs), systolicBloodPressure(?vs, ?x), float[>= 130.0f](?x)	hasSystolicPressureOutOfExpectedRange(?vs, true)	Infer Field Value
R4	VitalSigns(?vs), systolicBloodPressure(?vs, ?x), float[<= 100.0f](?x)		
R5	VitalSigns(?vs), isActive(?vs, true), hasSystolicPressureOutOfExpectedRange(?vs, true), CardiacFunction(?cf)	position(?cf, 99)	Decrease Form Priority
R6	VitalSigns(?vs), hasSystolicPressureOutOfExpectedRange(?vs, true), isActive(?vs, true), CardiacFunction(?cf), position(?vs, ?po)	position(?cf, ?po)	Increase Form Priority
R7	ValuesAndBeliefsForm(?va), Evaluation(?e), hasPatient(?e, ?p), isActive(?e, true), hasAge(?p, ?age), integer[<= 12](?age)	isRemoved(?va, true)	Remove Form
R8	ValuesAndBeliefsForm(?va), Evaluation(?e), hasPatient(?e, ?p), isActive(?e, true), hasAge(?p, ?age), integer[>12](?age), originalPosition(?va, ?opo)	isRemoved(?va, false), position(?va, ?opo)	Include Form
R9	Evaluation(?e), Nutrition(?n), VitalSigns(?vs), float[¡= 18.0f](?bmi), isActive(?vs, true), bmi(?vs, ?bmi), position(?vs, ?po)	position(?n, ?po)	Increase Form Priority
R10	Evaluation(?e), DevelopmentAndGrowth(?dg), VitalSigns(?vs), float[¡= 18.0f](?bmi), integer[¡= 18](?age), isActive(?vs, true), age(?e, ?age), bmi(?vs, ?bmi), position(?vs, ?po)	position(?gd, ?po)	Increase Form Priority
R11	Nutrition(?n), VitalSigns(?vs), bmi(?vs, ?bmi)	bmi(?n, ?bmi)	Propagate Field Value

adaptations. The first one is the BMI value inferred from fields height and weight (Rule R2). The second observable adaptation is the order change of activities ‘Nutrition’ and ‘Development and Growth’. This occurs because Rules R9 and R10 identify that the patient is still in the growth phase and his BMI is below the expected level. The change of order can be seen comparing the lists at the top right of Figures 4.4 and 4.5.

These interactions show a few examples of our context-aware adaptable mechanism which allows to provide dynamic adaptations to aid health professionals in four aspects: change activ-

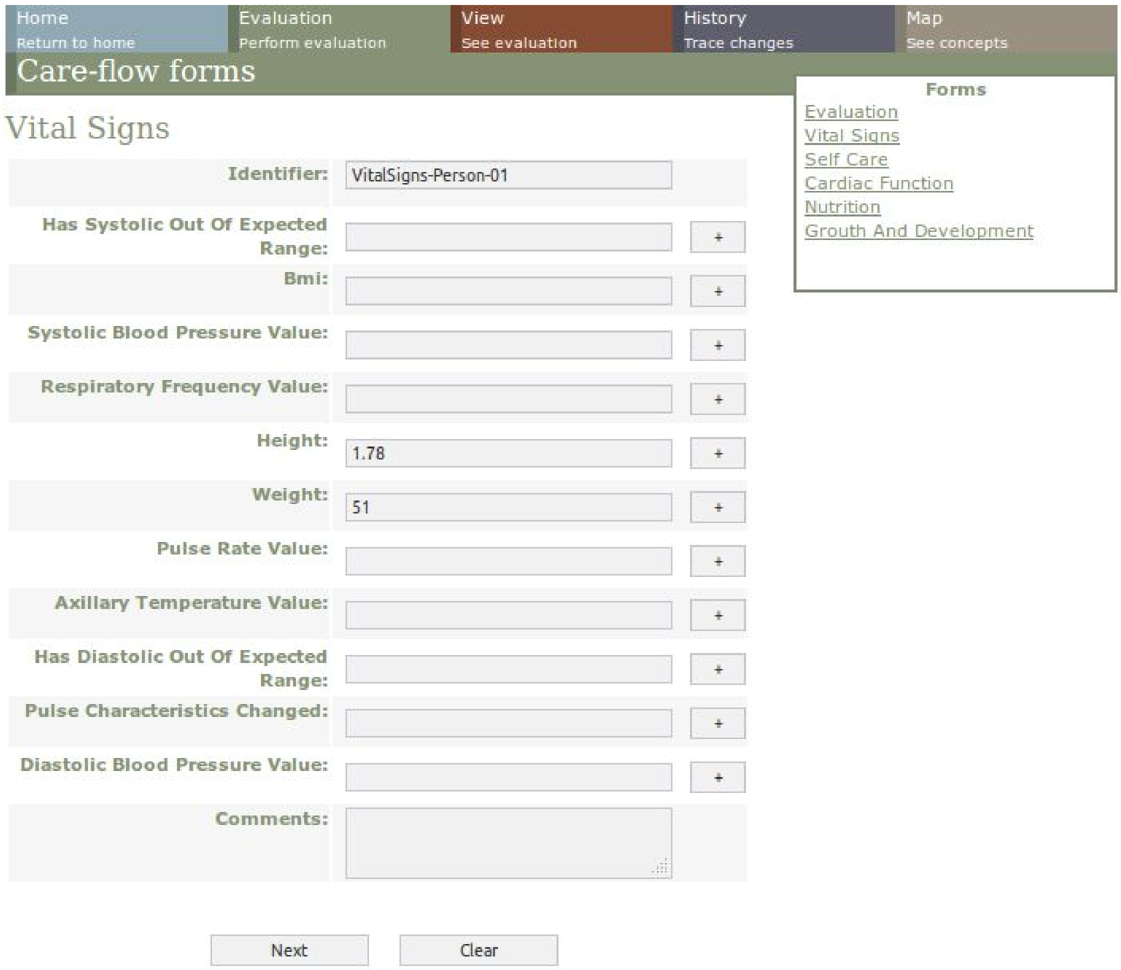

Figure 4.4: Screenshot presenting the Web Application interface before the form values are saved.

ity order, infer field values, avoid the repeated filling of values of fields by propagating values and include or exclude activities. On a single interaction it is possible to perform all adaptations along all activities.

4.4 The Context Adaptation Engine

The general algorithm performed by the *CAEng* is the following:

```
For each CRec received from Semantic Mapper do
 //Pre-processing CRec
 CRec.properties[Monitor.CurrentActivy].isActive = true
 For each Property p on CRec
   p.timestamp = now()

 //Inferring adaptations
 CRec' = Reasoner.reason(Ontology, CRec)
```

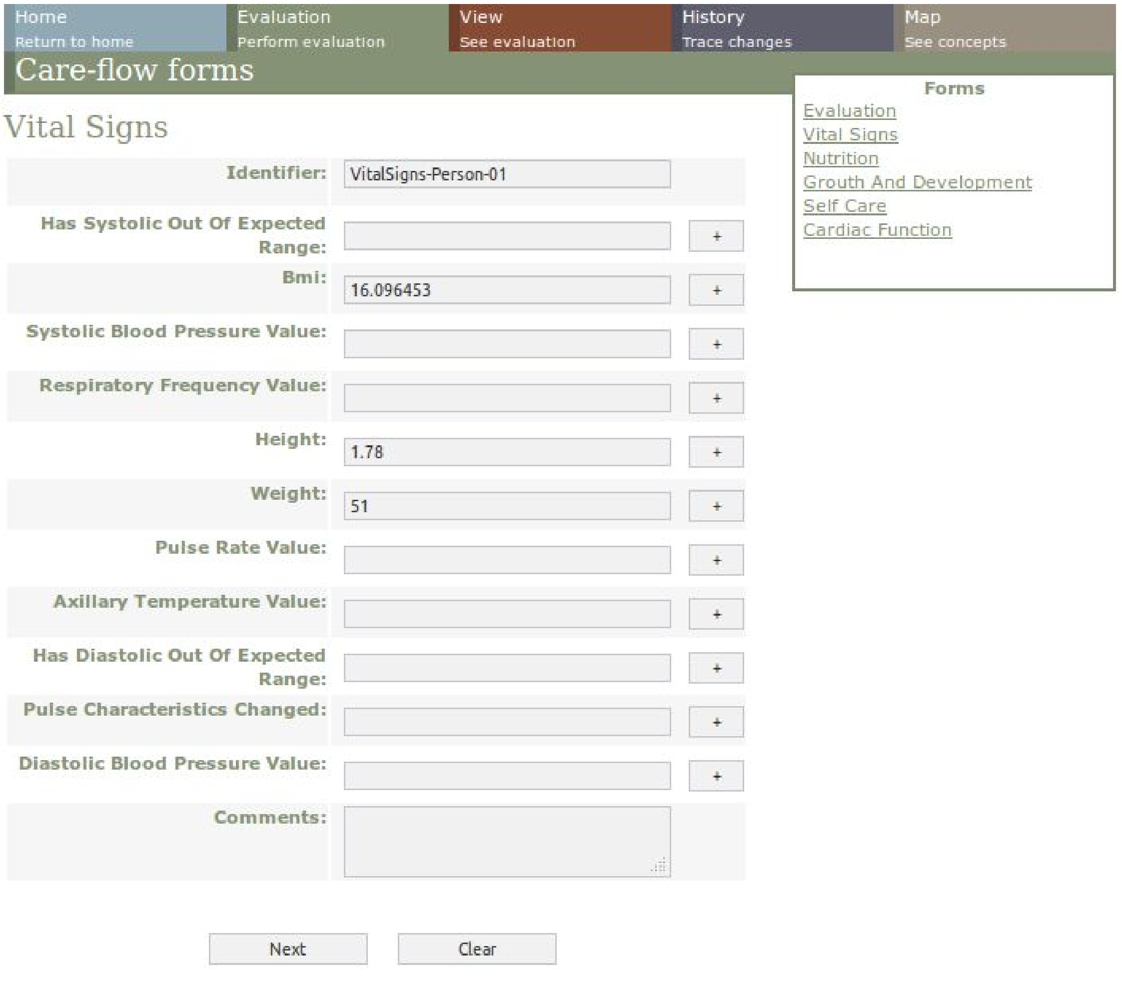

Figure 4.5: Screenshot presenting the Web Application interface after the form values are saved and all adaptation rules are applied.

```
//Post-processing CRec'
For each Property p on CRec'
   For each Property q on p.isPropertyOf.Properties
               if p.uri == q.uri
                                      if p.timestamp < q.timestamp
                                          CRec.remove(p)
                              else
                                          CRec.remove(q)
           if p.localName =  and p.value = true
      p.remove()
    CRec.properties[Monitor.CurrentActivy].isActive = false
```

The Ontology is instantiated for each new patient, and is progressively updated during workflow execution, according to changes in context. This instantiation reflects changes in the workflow.

The reasoning process generates a new version of *CRec* (*CRec'*). This version contains all

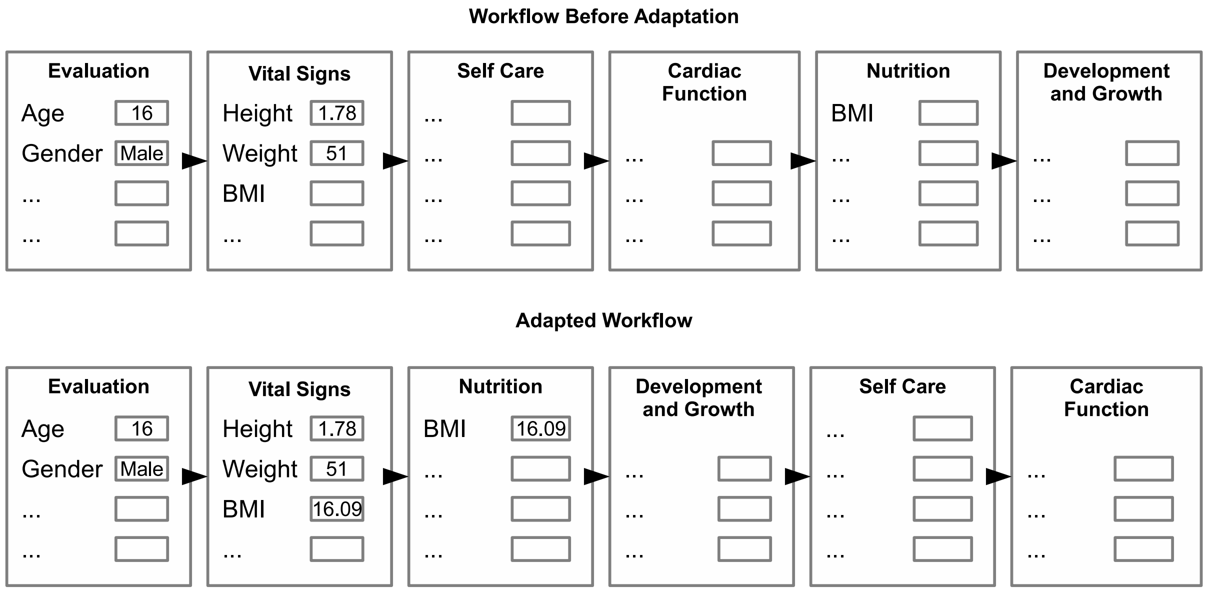

Figure 4.6: Workflow before (top) and after (bottom) adaptation. New values on bottom workflow are automatically propagated and/or inferred.

adaptations recommended according to the previous interaction. Timestamping avoids duplication of properties during reasoning.

Since a new workflow state is represented as an ontology instance, the new, recommended, worflow state can be retrieved using a SPARQL query:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX base:<http://kraho.org/ontologies/procenf.owl#>
SELECT DISTINCT ?task
WHERE {
 ?subject rdf:type ?task .
 ?subject base:originalPosition ?opo .
 ?subject base:position ?po .
 FILTER(?task != owl:NamedIndividual && ?task != owl:Class)
}
ORDER BY ?position ?opo
```

This query lists all activities from CREC' sorting them according to the presentation order defined. When two or more activities have the same presentation order, the original order (of the previous state) is used to prioritize them. So, even thought 'Nutrition' and 'Development and Growth' have the same presentation order, because of rules R9 and R10, 'Nutrition' will be shown first.

4.5 Related Work

Related work basically concerns dynamic adaptation of plans in health systems. Such systems are called CIG (Computer-Interpretable Guidelines), an evolution of CPG (Clinical Practice Guidelines) [33]. Most implementations and models follow the paradigm of TNM [34], while we adapt workflow systems to incorporate the advantages of TNM, as discussed in this section.

Dynamic workflow adaptation is originally found in the context of business environments (e.g., [42]). However, for other situations, and particularly healthcare, context-sensitive adaptation has not been explored. This section compares our proposal with related studies considering such issues, in particular showing the theory behind our architectural choices, for healthcare applications.

In previous work ([78]) we compared (Business and Scientific) Workflows and TNM approaches. As a result, we proposed the notion of context-aware workflow, presented on Section 4.2, inspired on TNM. Here, we detail prominent TNM characteristics and how they were incorporated in our work.

4.5.1 Healthcare scenario and adoption of TNM over other approaches

We analyzed TNM because it succeeds over other approaches proposed as solutions to be used on CIGs. According to [57], alternative proposals such as MLMs and ADTs do not provide full support for conceptualizing a multi-step guideline that unfolds over time. Also, a TNM can model alternative pathways and sequences of tasks, unlike rule-based systems. Thus, they are an appropriate choice to combine with SWfMS.

According to Miksch and Shahar [52], traditional plan-execution representations have limitations that make them not applicable in dynamic changing environments like healthcare – e.g., they assume instantaneous actions and effects; and a goal may not be achievable. They also explain that the medical domain requires the support of temporal annotations, a rich set of parallel, sequential, and iterative operators, with well defined semantics for both prescribed actions and task-specific annotations. Considering those aspects, the authors proposed the ASBRU language, one of the most prominent TNMs. We point out that workflows support these and other operators, and that several SWfMS support activity annotation.

4.5.2 Keep rationale, knowledge and terminology of (healthcare) professionals

Usability is a key factor in systems. This aspect leads to two goals: systems must use appropriate terminology, and keep the rationale of the work procedures of given user profiles.

TNMs achieve these goals by adopting content from clinical guidelines. As Peleg et al. [57] explain, TNMs define abstractions that help in conceptualizing guideline logic and in interpreting data. To Ye et al. [84], a model to represent a clinical pathway needs to explicitly represent the knowledge about outcomes, resources, variances and time aspects in order to determine, monitor and evaluate the performance of staff activities and the quality of patient care. The need for a connection between plans and concepts was also a concern of As Din et al. [29]. They created a methodology in which elements of care plans, already personalized for a patient, are associated with elements from nursing clinical practice guidelines. These authors argue that quality and standardization are ensured as result of the association between care plan and guidelines, .

The relation between plan to be executed, terminology, concepts, and methodology allows to achieve another important aspect cited by [27]: creation of support tools for clinical decision to "make it easy to do it right". This is being made possible due to the development of systems that reduce the complexity to deal with large amounts of data and refine aggregate information to obtain results that suit the case.

This concept of making a system familiar to the professional and which provides reliable aid is important for different domains. Adapting workflows to professional methodologies and standards improves the change of acceptance. We keep the rationale, knowledge and terminology of nursing professionals by adopting ontologies to describe the domain vocabulary and semantics, and SWRL rules to embed practices and knowledge. As concepts and rules are human readable and computer interpretable, it is possible to make translations that makes easier to the domain specialists to create, understand and change the elements described.

4.5.3 Hierarchical organization of tasks

To preserve the methodology of healthcare professionals, one should also keep the organization from original clinical guidelines. Burgers et al. [15] show that clinical guidelines are developed in a top-down process. Organizations use evidences and discuss about practices that are effective in a treatment. Since the evidences are supported by a range of data, there are some specificities that may not be present in guidelines, but are partially covered.

Since TNMs are developed focusing on healthcare professionals, they are close to the methodology of clinical guidelines. As [57] observed, "groups have adopted different approaches reflecting their interests and expertise, many of the approaches have in common a hierarchical decomposition of guidelines into networks of component tasks that unfold over time". This occurs because CIGs "must be able to represent various kinds of guidelines that may differ considerably in complexity and level of abstraction, for example by means of nesting or decomposition" [24]. Also, TNMs allow to express various arrangements of these components and interrelationships between them" [57].

Examples of hierarchical models in healthcare appear in Seyfang et al. [67], Dang et al. [22], and Din et al. [29]. While the two first are focused on the medical domain, the third is contextualized on nursing, reinforcing the evidence of the hierarchical organization for healthcare.

In our architecture, we create the notion hierarchical navigation by adding rules that allow to perform alternative paths through the workflow activities. Instead of having a serial and fixed execution of activities, it is possible to skip or change activity order.

4.5.4 Arguments for decision: non-deterministic arguments and contextual elements

Din et al. [29] stated that in "patient care needs to be both customized to the patient's care needs and standardized in terms of best evidence". TNMs have both: a) the knowledge from different specialists, resulting from several cases, that can be selected to treat patients and achieve a goal. b) the specific information about the patient that will be used as argument for a decision;

Knowledge can be extracted from CPGs, which represent clinicians' background about suggested ways to deal with health issues. However, given the high amount of information present in guidelines, there is a need to facilitate and speed up the identification of the specific guideline suitable for the situation. TNMs found in the literature provide three elements to deal with this requirement: a) hierarchical organization, b) deterministic and non-deterministic arguments to select specific plans, and c) contextual elements. The first one was already explained on Section 4.5.3. The second and the third ones combine elements that describe specificities of the case, considering patient data, guideline arguments and elements that characterize the treatment.

As presented in Peleg et al. [57], TNMs allow, by formal expressions, to define abstract terms to represent data. For example, "isolated systolic hypertension" is an abstract representation of the situation in which "patients not taking anti-hypertensive agents have systolic blood pressures of at least 140 mm Hg and diastolic blood pressures of less than 90 mm Hg.' Also, there are TNMs that support context variables – Peleg et al. [57] explains that GLIF and EON provide a Scenario to support context description, while GLIF has Patient State.

4.6 Conclusions and Future Work

This paper presented DynFlow – our architecture to adapt scientific workflows to a given context, and instantiated it for care-flow situations. As a result, we increased the flexibility of SWfMS rendering them amenable to on-the-fly context sensitive adaptation. DynFlow is heavily centered on ontologies and adaptation rules. Those rules reflect domain knowledge, and allow the use of domain specific terms to determine which adaptations should be made to a workflow for a given context (i.e., state of the patient under treatment). The inspiration from CIGs was convenient to put SWfMS closer to the domain of healthcare, given that those systems are suitable for dynamic scenarios

Future work includes interface improvement and enhancing the set of adaptation rules to aid healthcare professionals. Interface changes include: I) a new kind of visualization that shows a hierarchical representation of workflows, according to choices made using adaptation rules; II) search mechanisms using filtering criteria over ontology concepts. Also, other care-flow steps can be supported, for instance, by creating rules to suggest possible diseases during the diagnostics phase.

Chapter 5

Conclusions and Extensions

In this thesis we addressed some issues concerning the challenges involved in dynamically customizing scientific workflows based on context. In spite of extensive work on workflows and WfMS, relatively little has been done in this subject, in particular for scientific workflows. From another point of view, this work can also be analyzed under the perspective of Decision Support Systems.

This is also an issue for less dynamic scenarios of use. Dynamic adaptation helps the efficient use of workflows with respect to important features: easy understanding by non computer specialists, reuse of components, design adaptation and flexibility to explore a wide range of parameter possibilities. For instance, for each variable included in an experiment, new conditions may be applied, creating new versions of the workflow. The multiplication of workflows may become more complex to manage.

5.1 Main Contributions

Our first contribution is presented on Chapter 2. We studied systems that deal with context changes and analyzed scientific workflow and healthcare systems. The analysis performed in Chapter 2 allowed us to argue that workflow and healthcare systems have complementary features. This led us to characterize two main groups of features: infrastructure and organization. While the first group includes features such as provenance, extensibility and reuse, the second group includes among others, the change of the flow during execution, organization of the basic elements and context modeling. This characterization showed us that workflow systems usually provide more infrastructure features, while healthcare systems have a more sophisticated set of organization features that allows to provide a more dynamic execution. Moreover, we point out that the effort to provide (workflow-like) infrastructure features into healthcare systems is much larger than the effort to provide (healthcare-like) organization features into workflow systems. This observation led us to the study, presented in Chapter 3, of healthcare planning systems characterized as CIG. Among them, we identified TNMs as the most prominent.

Our second and main contribution is the combination of workflow systems and TNM models. The blend of both allows to keep infrastructure features from a SWfMS and perform adaptations on the fly from TNM. This resulted in the specification of our architecture, briefly presented in Chapter 3.

Our third contribution is the full specification of the architecture with description of its implementation in a Web based prototype, presented in Chapter 4. This prototype allowed us to run a case study in nursing care planning, using real data. The prototype implemented our architecture hiding the workflow concepts from users to make it more natural to be used.

The DynFlow architecture can be used with different workflow systems and can be applied to domains different from nursing. Each workflow system requires customization of the Context Adaptation Layer. For each application domain, new Adaptation Rules, Ontology and CRec should be created. Thus, for each application our research is not restricted to a single workflow system nor to a single domain.

Our work contributes both to research in Computer Science and the application domain. For Computer Science, it points out how to blend TNM into SWfMS. From the healthcare point of view, this is a domain which needs context-aware adaptation and where researchers are not aware of scientific workflow systems.

However, we observe it still presents limitations such as the need for computing experts to translate end user requests into rules. Furthermore, there is a need for extensive validation from end users. Other issues and extensions are briefly enumerated next.

5.2 Extensions

This work can be continued in different aspects:

- Develop a graphical user interface that incorporates additional TNM characteristics: There are TNM characteristics that can be better supported by creating a graphical user interface to better present information. For instance, displaying the hierarchical organization of tasks, created by the adaptation rules. Thus, all unfolding tasks that may occur over time can be seen and interacted with.

- Use provenance to optimize adaptations: It is possible to analyze the provenance of a workflow execution to perform additional customization by creating new adaptation rules. For example, choosing tasks according to the smallest execution time or to the best results, or even suggesting tasks to the user.

- Extend the Web Prototype to act on additional steps of nursing care, such as diagnosis and interventions. This requires associating more information about context and domain to allow adaptations based on actions that may occur on seemingly unrelated future steps.

- Investigate performance optimization to perform adaptations using less computational resources and within a reduced execution time.
- Investigate the use of workflow patterns, such as those provided by Russell et al. [62] and Aghakasiri et al. [1] to model and dynamically adapt workflows.
- Create interfaces to allow healthcare users to insert domain rules. Additional interface support can show users the reasoning process, to help decision support.
- Investigate the proposal in the context of Decision Support Systems and Autonomous Agents. Furthermore, the same work can be contrasted with self adaptive systems of Software Engineering. In particular, our prototype implementation does not contemplate self adaptation (e.g., invoking new modules). However, rules can be used to provide other kinds of adaptations.

Bibliography

[1] Z. Aghakasiri and S.-H. Mirian-Hosseinabadi. Workflow change patterns: Opportunities for extension and reuse. In R. Lee and N. Ishii, editors, *Software Engineering Research, Management and Applications 2009*, volume 253 of *Studies in Computational Intelligence*, pages 265–275. Springer Berlin Heidelberg, 2009.

[2] W. Aigner and S. Miksch. Carevis: Integrated visualization of computerized protocols and temporal patient data. *Artificial Intelligence in Medicine*, 37(3):203–218, 2006.

[3] I. Altintas, C. Berkley, E. Jaeger, M. Jones, B. Ludascher, and S. Mock. Kepler: an extensible system for design and execution of scientific workflows. In *Proc. of 16th Intl. Conf. on Scientific and Statistical Database Management, 2004.*, pages 423–424. Ieee, 2004.

[4] J. F. Arocha, D. Wang, and V. L. Patel. Identifying reasoning strategies in medical decision making: A methodological guide. *Journal of Biomedical Informatics*, 38(2):154 – 171, 2005.

[5] O. Asfari, B. Doan, Y. Bourda, and J. Sansonnet. A Context-based Model for Web Query Reformulation. *limsi.fr*, 2010.

[6] S.-J. Baek, J.-S. Han, and K.-Y. Chung. Dynamic reconfiguration based on goal-scenario by adaptation strategy. *Wireless Personal Communications*, 73(2):309–318, 2013.

[7] K. Bandara, M. Wang, and C. Pahl. Dynamic Integration of Context Model Constraints in Web Service Processes. In *Software Engineering*. ACTA Press, 2009.

[8] J. E. Bardram and T. R. Hansen. Context-Based Workplace Awareness. *Computer Supported Cooperative Work (CSCW)*, 19(2):105–138, Mar. 2010.

[9] R. Barga, J. Jackson, N. Araujo, D. Guo, N. Gautam, K. Grochow, and E. Lazowska. Trident: Scientific workflow workbench for oceanography. In *Services - Part I, 2008. IEEE Congress on*, pages 465–466, july 2008.

[10] P. Barthelmess. Sistemas de workflow: Análise da área e proposta de modelo. Master's thesis, Instituto de Computação - Unicamp, Junho 1996.

[11] L. S. Baumann and R. D. Coop. Automated workflow control: A key to office productivity. In *Proceedings of the May 19-22, 1980, National Computer Conference*, AFIPS '80, pages 549–554, New York, NY, USA, 1980. ACM.

[12] M. Beyer, K. A. Kuhn, C. Meiler, S. Jablonski, and R. Lenz. Towards a flexible, process-oriented it architecture for an integrated healthcare network. In *Proceedings of the 2004 ACM Symposium on Applied Computing*, SAC '04, pages 264–271, New York, NY, USA, 2004. ACM.

[13] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, and K. Vahi. Characterization of scientific workflows. *2008 Third Workshop on Workflows in Support of Large-Scale Science*, pages 1–10, Nov. 2008.

[14] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca. A data-oriented survey of context models. *SIGMOD Rec.*, 36:19–26, December 2007.

[15] J. S. Burgers, R. Grol, N. S. Klazinga, M. Makela, and J. Zaat. Towards evidence-based clinical practice: an intl. survey of 18 clinical guideline programs. *Intl. Journal for Quality in Health Care*, 15(1):31–45, Feb. 2003.

[16] F. Carmagnola, F. Cena, C. Gena, and I. Torre. A semantic framework for adaptive web-based systems. In *Proc. of SWAP*, pages 1613–0073. Citeseer, 2005.

[17] J. Cheng, S. Nara, and Y. Goto. Freeencal: A forward reasoning engine with general-purpose. In B. Apolloni, R. Howlett, and L. Jain, editors, *Knowledge-Based Intelligent Information and Engineering Systems*, volume 4693 of *Lecture Notes in Computer Science*, pages 444–452. Springer Berlin Heidelberg, 2007.

[18] Y. Cho, J. Moon, and H. Yoe. LNCS 6018 - A Context-Aware Service Model Based on Workflows for u-Agriculture. *Technology*, pages 258–268, 2010.

[19] CNC. Cnc - overview: Nursing outcomes classification (noc). 2012.

[20] P. Dadam and M. Reichert. The adept project: a decade of research and development for robust and flexible process support. *Computer Science - Research and Development*, 23(2):81–97, 2009.

[21] J. Dallien, W. MacCaull, and A. Tien. Initial work in the design and development of verifiable workflow management systems and some applications to health care. In *Proceedings of the 2008 5th International Workshop on Model-based Methodologies for Pervasive and*

Embedded Software, MOMPES '08, pages 78–91, Washington, DC, USA, 2008. IEEE Computer Society.

[22] J. Dang, A. Hedayati, K. Hampel, and C. Toklu. An ontological knowledge framework for adaptive medical workflow. *Journal of biomedical informatics*, 41(5):829–36, Oct. 2008.

[23] S. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *SIGMOD Conf.*, pages 1345–1350. Citeseer, 2008.

[24] P. de Clercq and K. Kaiser. Computer-interpretable guideline formalisms. *Studies in health technology*, pages 22–43, 2008.

[25] E. Deelman and A. Chervenak. Data Management Challenges of Data-Intensive Scientific Workflows. *2008 Eighth IEEE Intl. Symposium on Cluster Computing and the Grid (CCGRID)*, pages 687–692, May 2008.

[26] E. Deelman, D. Gannon, M. Shields, and I. Taylor. Workflows and e-Science: An overview of workflow system features and capabilities. *Future Generation Computer Systems*, 25(5):528–540, May 2009.

[27] P. R. Dexter, S. Perkins, J. M. Overhage, K. Maharry, R. B. Kohler, and C. J. McDonald. A computerized reminder system to increase the use of preventive care for hospitalized patients. *New England Journal of Medicine*, 345(13):965–970, 2001. PMID: 11575289.

[28] A. K. Dey. *Providing architectural support for building context-aware applications*. PhD thesis, Georgia Tech, Atlanta, GA, USA, 2000. AAI9994400.

[29] M. A. Din, S. S. R. Abidi, and B. Jafarpour. Ontology based modeling and execution of Nursing Care Plans and Practice Guidelines. *Studies in health technology and informatics*, 160(Pt 2):1104–8, Jan. 2010.

[30] M. Doenges and M. Moorhouse. *Application of nursing process and nursing diagnosis: an interactive text for diagnostic reasoning*. G - Reference, Information and Interdisciplinary Subjects Series. F.A. Davis, 2008.

[31] A. El-Fakdi, J. Meléndez, and R. Petite. A proposal of a workflow-based cbr for interventional cdss - application to transcatheter aortic valve implantation (tavi). In *HEALTHINF*, pages 201–206, 2012.

[32] J. Fox, V. Patkar, I. Chronakis, and R. Begent. From practice guidelines to clinical decision support: closing the loop. *Journal of the Royal Society of Medicine*, 102(11):464–73, Nov. 2009.

[33] A. Gonzalez-Ferrer, A. ten Teije, J. Fdez-Olivares, and K. Milian. Automated generation of patient-tailored electronic care pathways by translating computer-interpretable guidelines into hierarchical task networks. *Artificial Intelligence in Medicine*, 2012 (in print).

[34] P. Gooch and A. Roudsari. Computerization of workflows, guidelines, and care pathways: a review of implementation challenges for process-oriented health information systems. *Journal of the American Medical Informatics Association : JAMIA*, 18(6):738–48, 2011.

[35] S. Holl, O. Zimmermann, and M. Hofmann-Apitius. A new optimization phase for scientific workflow management systems. In *Proceedings of the 2012 IEEE 8th International Conference on E-Science (e-Science)*, E-SCIENCE '12, pages 1–8, Washington, DC, USA, 2012. IEEE Computer Society.

[36] J. H. Holland. Complex Adaptive Systems. *Daedalus*, 121(1):pp. 17–30, 1992.

[37] D. Hollingsworth and Others. Workflow management coalition: The workflow reference model. *Workflow Management Coalition*, (1):1–55, 1993.

[38] B. Howe, P. Lawson, R. Bellinger, J. Freire, E. Anderson, E. Santos, C. Scheidegger, A. Baptista, and C. Silva". End-to-end escience: Integrating workflow, query, visualization, and provenance at an ocean observatory. In *Proc. of the 2008 Fourth IEEE Intl. Conf. on eScience"*, pages 127–134, 2008.

[39] F.-S. Hsieh and J.-B. Lin. Development of context-aware workflow systems based on petri net markup language. *Computer Standards & Interfaces*, 36(3):672 – 685, 2014.

[40] D. Hull, K. Wolstencroft, R. Stevens, C. Goble, M. R. Pocock, P. Li, and T. Oinn. Taverna: a tool for building and running workflows of services. *Nucleic acids research*, 34(-):W729–32, July 2006.

[41] N. Intl. *Nursing Diagnoses: Definitions and Classification 2012-14 (NANDA Intl. Nursing Diagnosis)*. Wiley, 2012.

[42] P. J. Kammer, G. A. Bolcer, R. N. Taylor, A. S. Hitomi, and M. Bergman. Techniques for supporting dynamic and adaptive workflow. *Computer Supported Cooperative Work*, 9(3/4):269–292, 2000.

[43] M. Kholladi and B. Doan. PRESY: A Context Based Query Reformulation Tool for Information Retrieval on the Web. *Annals of Physics*, 54(4):258, 2010.

[44] T. Y. Leong, K. Kaiser, and S. Miksch. Free and open source enabling technologies for patient-centric, guideline-based clinical decision support: a survey. *Yearbook of medical informatics*, (April):74–86, Jan. 2007.

[45] B. Ludäscher, I. Altintas, S. Bowers, J. Cummings, T. Critchlow, E. Deelman, D. D. Roure, J. Freire, C. Goble, M. Jones, S. Klasky, T. McPhillips, N. Podhorszki, C. Silva, I. Taylor, and M. Vouk. Scientific process automation and workflow management. In A. Shoshani and D. Rotem, editors, *Scientific Data Management*, Computational Science Series, chapter 13. Chapman & Hall, 2009.

[46] J. Lundberg. Principles of workflow support in life critical situations. *Licentiate thesis, School of Engineering, Blekinge Institute of Technology*, 2007.

[47] J. Lundberg and G. Rune. Robust approach towards context dependant information sharing in distributed environments. *ICEIS09, Milan, Italy*, 2009.

[48] D. Martin, M. Burstein, J. Hobbs, O. Lassila, D. McDermott, S. McIlraith, S. Narayanan, M. Paolucci, B. Parsia, T. Payne, E. Sirin, N. Srinivasan, and K. Sycara. OWL-S: Semantic Markup for Web Services – 1.2 Release, 2008.

[49] C. B. Medeiros, J. Pérez-Alcazar, L. Digiampietri, G. Z. P. Jr., A. Santanchè, R. da Silva Torres, E. Madeira, and E. Bacarin. Woodss and the web: Annotating and reusing scientific workflows. *SIGMOD Record*, 34(3):18–23, 2005.

[50] B. Mikolajczak and M. Shenoy. Flexibility through case handling in careflow systems: A case study of cutaneous melanoma. In *Health Care Management (WHCM), 2010 IEEE Workshop on*, pages 1 –6, feb. 2010.

[51] S. Miksch, R. Kosara, Y. Shahar, and P. D. Johnson. AsbruView: Visualization of Time-Oriented, Skeletal Plans. pages 11–18, 1998.

[52] S. Miksch, Y. Shahar, and P. Johnson. Asbru: A task-specific, intention-based, and time-oriented language for representing skeletal plans. In *UK, Open University*, pages 9–1, 1997.

[53] I. C. Millard, D. D. Roure, and N. Shadbolt. The use of ontologies in contextually aware environments. In J. Indulska and D. D. Roure, editors, *Proc. of First Intl. Workshop on Advanced Context Modelling, Reasoning And Management*, pages 42–47, Nottingham, UK, 2004.

[54] E. Ogasawara, L. Murta, C. Werner, and M. Mattoso. Linhas de Experimento: Reutilização e Gerência de Configuração em Workflows Científicos. In *2 E-Science Workshop co-locado ao SBBD/SBES*, pages 31–40, 2008.

[55] E. Ogasawara, C. Paulino, L. Murta, C. Werner, and M. Mattoso. Experiment line: Software reuse in scientific workflows. In M. Winslett, editor, *Scientific and Statistical*

Database Management, volume 5566 of *Lecture Notes in Computer Science*, pages 264–272. Springer Berlin Heidelberg, 2009.

[56] S. Panzarasa and M. Stefanelli. Workflow management systems for guideline implementation. *Neurological sciences : official journal of the Italian Neurological Society and of the Italian Society of Clinical Neurophysiology*, 27 Suppl 3:S245–9, June 2006.

[57] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R. A. Greenes, S. Miksch, S. Quaglini, A. Seyfang, E. H. Shortliffe, M. Stefanelli, and et al. Comparing computer-interpretable guideline models: A case-study approach. *JAMIA*, 10:2003, 2003.

[58] H. H. C. Peres, D. D. A. L. M. D. Cruz, A. F. C. Lima, R. R. Gaidzinski, D. C. F. Ortiz, M. M. E. Trindade, R. Tsukamoto, and N. B. Conceição. Desenvolvimento de Sistema Eletrônico de Documentação Clínica de Enfermagem estruturado em diagnósticos, resultados e intervenções. *Revista da Escola de Enfermagem da USP*, 43(spe2):1149–1155, Dec. 2009.

[59] M. Reichert and B. Weber. Process change patterns: Recent research, use cases, research directions. In *Seminal Contributions to Information Systems Engineering*, pages 397–404. Springer Berlin Heidelberg, 2013.

[60] D. Riaño, F. Real, J. A. López-Vallverdú, F. Campana, S. Ercolani, P. Mecocci, R. Annicchiarico, and C. Caltagirone. An ontology-based personalization of health-care knowledge to support clinical decisions for chronically ill patients. *Journal of biomedical informatics*, Jan. 2012.

[61] S. Rinderle, M. Reichert, and P. Dadam. Correctness criteria for dynamic changes in workflow systems??a survey. *Data & Knowledge Engineering*, 50(1):9 – 34, 2004. Advances in business process management.

[62] N. Russell, A. H. M. ter Hofstede, D. Edmond, and W. M. P. van der Aalst. Workflow data patterns: Identification, representation and tool support. In *Proceedings of the 24th International Conference on Conceptual Modeling*, ER'05, pages 353–368, Berlin, Heidelberg, 2005. Springer-Verlag.

[63] S. W. Sadiq, M. E. Orlowska, and W. Sadiq. Specification and validation of process constraints for flexible workflows. *Information Systems*, 30(5):349 – 378, 2005.

[64] C. E. Scheidegger, H. T. Vo, D. Koop, J. Freire, and C. T. Silva. Querying and re-using workflows with VsTrails. *Proc. of the 2008 ACM SIGMOD intl. conf. on Management of data - SIGMOD '08*, page 1251, 2008.

[65] S. Schick, H. Meyer, M. Bandt, and A. Heuer. Enabling yawl to handle dynamic operating room management. In F. Daniel, K. Barkaoui, and S. Dustdar, editors, *Business Process Management Workshops (2)*, volume 100 of *Lecture Notes in Business Information Processing*, pages 249–260. Springer, 2012.

[66] H. Schonenberg, R. Mans, N. Russell, N. Mulyar, and W. Aalst. Process flexibility: A survey of contemporary approaches. In J. Dietz, A. Albani, and J. Barjis, editors, *Advances in Enterprise Engineering I*, volume 10 of *Lecture Notes in Business Information Processing*, pages 16–30. Springer Berlin Heidelberg, 2008.

[67] A. Seyfang, S. Miksch, and M. Marcos. Combining diagnosis and treatment using ASBRU. *Intl. journal of medical informatics*, 68(1-3):49–57, Dec. 2002.

[68] Y. Shahar, S. Miksch, and P. Johnson. The asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine*, 14(1?2):29 – 51, 1998.

[69] L. M. Sigsby and D. W. Campbell. Nursing interventions classification: A content analysis of nursing activities in public schools. *Journal of Community Health Nursing*, 12(4):229–237, 1995.

[70] S. Smanchat, S. Ling, and M. Indrawan. A survey on context-aware workflow adaptations. In *Proceedings of the 6th International Conference on Advances in Mobile Computing and Multimedia*, MoMM '08, pages 414–417, New York, NY, USA, 2008. ACM.

[71] M. Sonntag, D. Karastoyanova, and E. Deelman. Bridging the gap between business and scientific workflows: Humans in the loop of scientific workflows. In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, pages 206–213, 2010.

[72] M. V. B. Souza and I. Augustin. Inferência de Atividades Clínicas a partir de Propriedades do Contexto. *Interface*, pages 1717–1726, 2010.

[73] T. Strang, C. Linnhoff-popien, and K. Frank. CoOL: A Context Ontology Language to Enable Contextual Interoperability. *Ifip Intl. Federation For Information Processing*, pages 236–247, 2003.

[74] L. H. THOM, M. REICHERT, C. IOCHPE, and J. P. OLIVEIRA. Why rigid process management technology hampers computerized support of healthcare processes. In *WIM - X Workshop de Informática Médica. Workshop do CSBC 2010.*, 2010.

[75] R. Turner. A model of explicit context representation and use for intelligent agents. *Modeling and Using Context*, pages 831–831, 1999.

[76] K. M. Unertl, M. B. Weinger, K. B. Johnson, and N. M. Lorenzi. Describing and modeling workflow and information flow in chronic disease care. *Journal of the American Medical Informatics Association : JAMIA*, 16(6):826–36, 2009.

[77] W. M. P. Van Der Aalst, A. H. M. Ter Hofstede, B. Kiepuszewski, and A. P. Barros. Workflow patterns. *Distrib. Parallel Databases*, 14(1):5–51, July 2003.

[78] B. S. C. M. Vilar, C. B. Medeiros, and A. Santanchè. Towards adapting scientific workflow systems to healthcare planning. In *HEALTHINF - International Conference on Health Informatics*, 2013.

[79] B. S. C. M. Vilar, C. B. Medeiros, and A. Santanchè. Dynflow - a software architecture for dynamic adaptation of workflows for healthcare planning. *Submmited*, 2014.

[80] B. S. C. M. Vilar, A. Santanchè, and C. B. Medeiros. Using healthcare planning features to drive scientific workflows on the web. In *WEBIST - International Conference on Web Information Systems and Technologies*, 2014.

[81] M. Z. Wei Tan. *Business and Scientific Workflows: A Web Service-Oriented Approach*. Wiley, March 2013.

[82] WfMC. Terminology and Glossary Document Number WFMC-TC-1011 - Issue 3.0. Technical report, Workflow Management Coalition, 1999.

[83] A. Yamin, I. Augustin, L. C. da Silva, R. A. Real, A. E. S. Filho, and C. F. R. Geyer. Exehda: adaptive middleware for building a pervasive grid environment. In *Proc. of the 2005 conf. on Self-Organization and Autonomic Informatics (I)*, pages 203–219, Amsterdam, The Netherlands, 2005. IOS Press.

[84] Y. Ye, Z. Jiang, X. Diao, D. Yang, and G. Du. An ontology-based hierarchical semantic modeling approach to clinical pathway workflows. *Comput. Biol. Med.*, 39(8):722–732, Aug. 2009.

[85] S. M. Yuval Shahar and P. Johnson. The Asgaard project: a task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial intelligence in medicine*, 14(1-2):29–51, 1998.