硕士学位论文

Dissertation for Master's Degree

(工程硕士)

(Master of Engineering)

基于DSL的家庭护理计划管理方法及工具设计与实现

DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS

吴元岭

哈爾濱工業大學　Université Blaise Pascal

2013 年 9 月

国内图书分类号：TP311 学校代码：10213
国际图书分类号：621.3 密级：公开

工程硕士学位论文
Dissertation for the Master’s Degree in Engineering

(工程硕士)

(Master of Engineering)

基于DSL的家庭护理计划管理方法及工具设计与实现

DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS

硕士研究生：吴元岭
导师：聂兰顺、副教授
副导师：Kun Mean HOU、教授
实习单位导师：Farouk Toumani、教授
申请学位：工程硕士
学科：软件工程
所在单位：软件学院
答辩日期：2013 年 9 月
授予学位单位：哈尔滨工业大学

Classified Index: TP311
U.D.C: 681

Dissertation for the Master’s Degree in Engineering

DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS

Candidate：	Yuanling Wu
Supervisor：	Lanshun Nie Associate Profeesor
Associate Supervisor:	Kun Mean HOU Proffessor
Industrial Supervisor:	Farouk Toumani Professor
Academic Degree Applied for：	Master of Engineering
Speciality：	Software Engineering
Affiliation：	School of Software
Date of Defense：	September, 2013
Degree-Conferring-Institution：	Harbin Institute of Technology

摘 要

家庭护理是指由医护人员在患者的家中提供卫生保健或支持性护理。近些年来，家庭护理正在广泛地开展起来，尤其是在人口老龄化严重的西方社会。同时，管理家庭护理计划面临严峻的挑战。

在一般情况下，传统管理家庭护理计划的方法无非是通过纸质文档记录在案或者使用业务流程建模（BPM）技术进行管理。然而，家庭护理计划可以看作是一种非结构化业务流程，因此它不适合使用在医疗领域中标准的业务流程建模技术，并且流程建模管理病不是一件容易的事。因此，我们提出了一种新型的基于 DSL（领域特定语言）的方法，通过使用高层次和面向用户的抽象来描述和管理家庭护理计划。考虑指定家庭护理计划的合理性，我们提出的领域特定语言是一种图形语言，它提供了适合家庭护理受益者的基本结构。首先通过从元模型到数据库的转换，我们获取稳定的家庭护理计划存储载体。基于此，我们使用 Vaadin 的框架来实现一种简化制定护理计划智能工具。最后，我们集合基于时间的自动机形式化描述领域特定语言来分析和描述初步结果以支持分析，验证和管理家庭护理计划。

该项目原型通过终端用户测试展示了可喜的成果。结果验证了基于DSL（领域特定语言）的方法是更合适和准确的来管理家庭护理计划。

关键词：领域特定语言，定时自动机，家庭护理计划; Vaadin 的框架

Abstract

In recent years, home care has been a real craze. It refers to health care or supportive care delivered by health care professionals in patients' homes, especially in the Western societies which are distributed to the aging population. Meanwhile, managing home care plans meets serious challenges. Our project aims to solve the problem how to manage home care plan.

In general, the traditional methods to manage home care plan are documents in the paper record or BPM (Business Process Modeling) technique. However, home care plans can be viewed as a kind of non-structured business processes which cannot be managed using standard BPM technology and process modeling in the medical field is in general not an easy task. Hence, we put up with a novel DSL (Domain Specific Language) based approach tailored to express home care plans using high level and user-oriented abstractions. Considering of reasonableness for specifying home care plan, the proposed DSL is a graphical language which provides basic constructs suitable for home care stakeholders. Through transformation from meta-model to rational database, we use vaadin framework to implement an intelligent tool to simplify the process of making care plan. Finally, we describe and discuss preliminary results regarding formalization of the proposed DSL abstractions using timed automata in order to provide basic services to support analysis, verification, enactment and management of home care plans.

The project prototype demonstrated promising results of the approach for the end user testing. The results validated the DSL (Domain Specific Language) based approach is more suitable and accurate to manage home care plans.

Keywords: Domain Specific Language; Timed Automata; Home Care Plan; Vaadin Framework

目 录

第 1 章 绪论

1.1 课题来源与背景

近年来，家庭护理 （也称为家居照顾) 已慢慢的流行开来。这是因为在西方社会，人口老龄化的管理越来越引起关注了。事实上，对于身体虚弱或者有慢性病的老人都非常需要持续和长期的照顾，所以家庭护理变得十分必要。家庭护理是指在病人的家中为其提供专业的卫生保健或支持性护理。出于社会和经济原因，家庭护理的目标是使需要照顾的人能够留在家中取代长期留在医院或卫生保健设施中。可以提供的几种类型的护理包括医疗服务如医院一级的护理和日常生活活动护理如洗澡、穿衣、搀扶去厕所等活动。这些服务由多学科的医疗团队制定，其中护理计划由一个给定的人或者专业的卫生保健协调人员所提供。这一计划通常是通过一个复杂的过程，涉及全面评估病人的需要（例如，医疗、护理、社会需要）以及其社会和物质环境构建的。

但是管理家庭护理计划面临很多难题。首先，在医疗领域中的过程建模不是一个简单的工作因为需要跨学科领域合作和来自不同医疗结构的参与者的协调工作。此外，家庭护理计划基本上都是非结构化的过程，也就是说，每个病人都有自己的特定保健计划。

在这种背景下，类似 Plas'O'Soin 都提出了用新型解决方案来管理家庭护理计划。Plas'O'Soin 是一个是由 ANR (Agence Nationale de la Recherche fran çaise) 和 ISIS 联合提出的实验项目。该项目的目标是提供一个平台以帮助监测和协调家庭护理。本课题即是 Plas'O'Soin 项目中的一个子项目。

1.2 课题意义

本课题“基于领域特定语言原理来管理家庭护理计划的方法的设计与实现”旨在设计一个提出一种新型的基于领域特定语言的方法来管理家庭护理计划来取代传统的管理策略。同时为医疗协调者开发出一个可视化的家庭护理计划工具来监督和管理家庭护理计划制定。我们的方法的关键在于对 DSL（领域特定语言），或者更具体的商业 DSL 的定义，使用高层次的抽象表达来量身制定家庭护理计划。我们提出 DSL 是一种图形语言，它提供了适合家庭保健利益相关者的基本构造。

1.3 国内外相关研究现状

1.3.1 家庭护理计划

护理计划是一个很广泛的词，主要是描述一个需要治疗一段较长时间的病人的临床情况的整个过程。它通常适用于慢性病照顾者或者涉及到很多学科团队综合涉及方案[7]。

在家庭护理计划，需要得到医疗护理的病人发送他们的要求到医院或社区等医疗机构。这些医疗机构将指定护士评估病人的情况，并提供相关的服务关怀计划列表。在病人接受该协议后，医生协调员开始做了详细的个人家庭护理计划包括行为，时间，进度等信息。一般的家庭护理计划过程如图 1-1 所示。

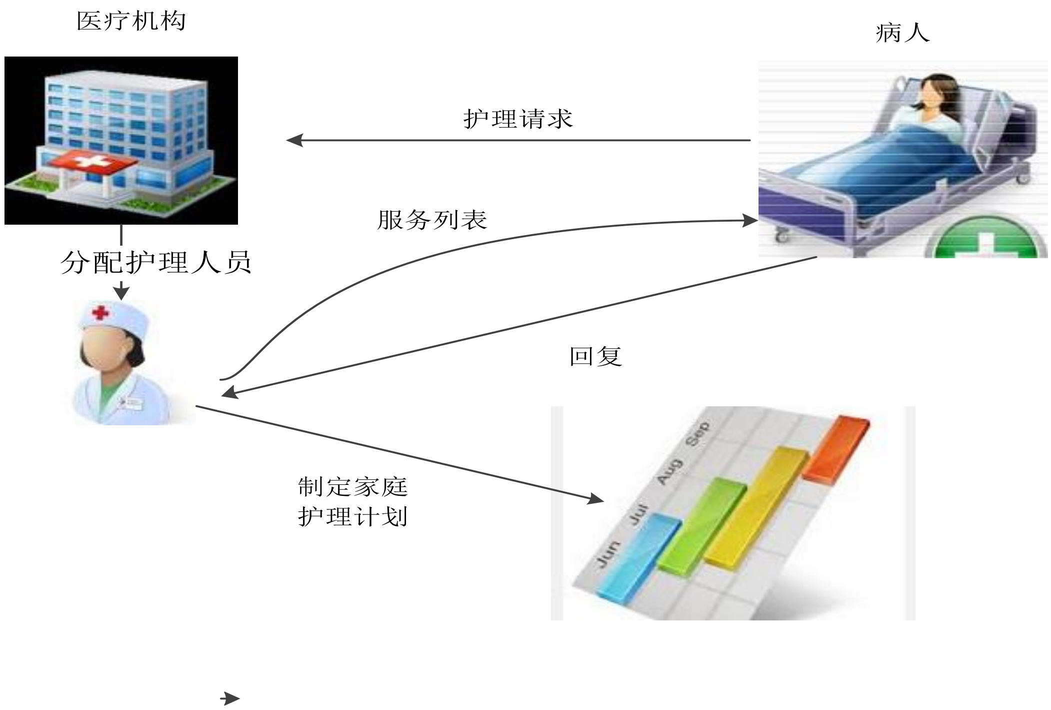
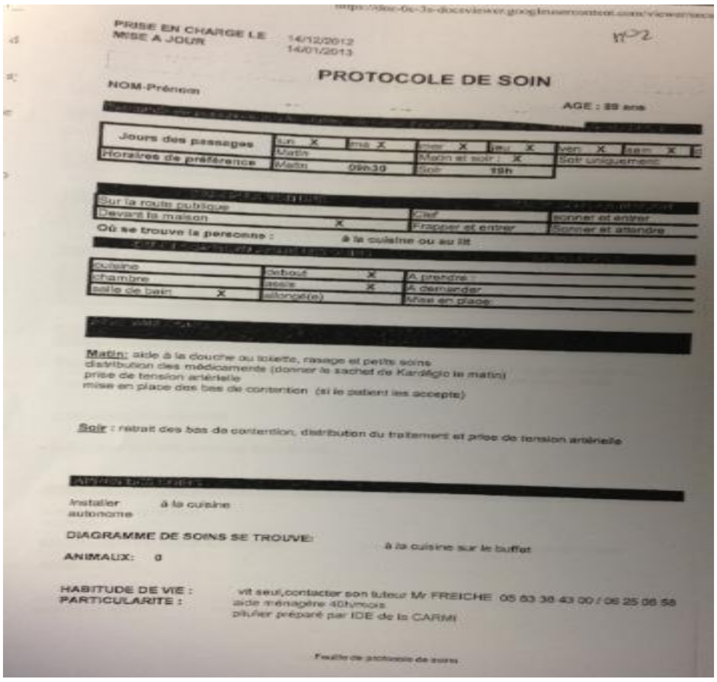


图 1-1 家庭护理计划过程

在正常情况下，计划管理人员（我们称之为计划协调者）习惯于使用纸质文档管理日常护理计划比如家庭管理计划文档或文件，如图 1-2 所示，即使这种方法在开始的步骤就要复杂得多。随着越来越多的患者接受家庭护理计划，护理计划也随之变得越来越多。这使得书面描述的家庭护理计划变得难以管理，比如调整或更新护理计划。特别地是，护理计划的文件往往缺乏统一的正式的结构，因此，协调员无法编辑出令他们满意的文档版本。更重要的是，对于那

些长期从事护理工作的人，他们积累的病人护理知识并无法体现在纸上。此外，如果信息不准确或不恰当的，它将是明显违背患者的安全性和持续护理计划。



PRISE EN CHARGE LE 14/12/2012
MISE A JOUR 14/01/2013

N°2

PROTOCOLE DE SOIN

NOM-Prénom

AGE : 89 ans

[illegible]

Jours des passages	lun X	ma X	mer X	jeu X	ven X	sam X	[illegible]
Horaires de préférence	Matin		Matin et soir : X		Soir uniquement		
	Matin	09h30	Soir	19h			

[illegible]

Sur la route publique			
Devant la maison	X	Clef	sonner et entrer
		Frapper et entrer	Sonner et attendre

Où se trouve la personne : à la cuisine ou au lit

[illegible]

cuisine				
chambre		debout	X	A prendre :
salle de bain	X	assis	X	A demander
		allongé(e)		Mise en place

[illegible]

Matin: aide à la douche ou toilette, rasage et petits soins
distribution des médicaments (donner le sachet de Kardégic le matin)
prise de tension artérielle
mise en place des bas de contention (si le patient les accepte)

Soir : retrait des bas de contention, distribution du traitement et prise de tension artérielle

[illegible]

Installer à la cuisine
autonome

DIAGRAMME DE SOINS SE TROUVE: à la cuisine sur le buffet

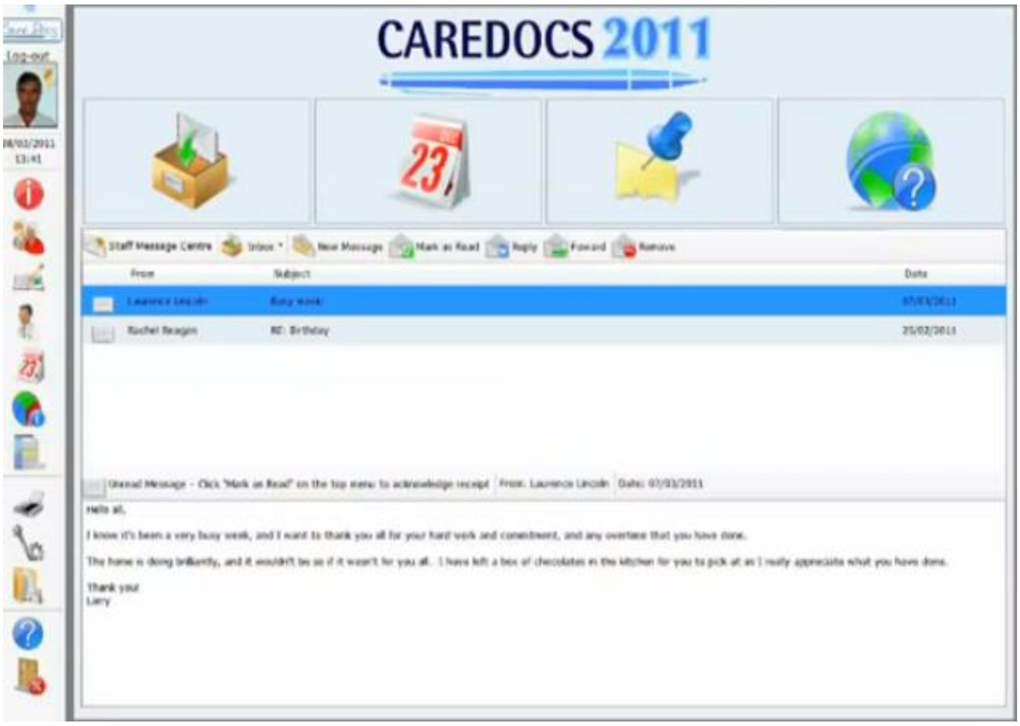
ANIMAUX: 0

HABITUDE DE VIE : vit seul,contacter son tuteur Mr FREICHE 05 53 38 43 00 / 06 25 08 58
PARTICULARITE : aide ménagère 40h/mois
pilulier préparé par IDE de la CARMI

Feuille de protocole de soins

图 1-2 家庭护理计划文档

为了解决上述我们所提到的问题，很多基于电脑的工具或者系统应运而生比如 CareDocs。CareDocs 是一个以计算机为基础的创新性的家庭护理计划管理软件。它的主要服务对象是室内和安老院护理。通过家庭护理者设计，CareDocs 提供所有的特性来保证家庭护理的平稳运行。护理计划是 CareDocs 的核心并且提供一个快速和容易的方式来进行全面的评估。这样就可以再短短几分钟内生产完全个性化的护理计划。管理者可以立刻获得一个关于居民，员工和护理计划信息的界面。然而，我们发现 CareDocs 系统只允许计划制定者输入信息代替纸质书写。这不能帮助计划制定者来制定计划和严重计划的合理性。CareDocs 见图 1-3。



From	Subject	Date
Laurence Lincoln	Busy week	07/03/2011
Rachel Reagan	RE: Birthday	25/02/2011

Unread Message - Click "Mark as Read" on the top menu to acknowledge receipt | From: Laurence Lincoln | Date: 07/03/2011

Hello all,

I know it's been a very busy week, and I want to thank you all for your hard work and commitment, and any overtime that you have done.

The home is doing brilliantly, and it wouldn't be so if it wasn't for you all. I have left a box of chocolates in the kitchen for you to pick at as I really appreciate what you have done.

Thank you!
Larry

图 1-3 现存的家庭护理计划软件 CareDocs

在很多好的病人管理信息系统中，网络基础设施和高效的沟通系统都是高效管理供应商之间护理计划的先决条件。在澳大利亚合作互利实验（第 1 轮：1997 年至 1999 年；第 2 轮：2002 年至今）中，一定数量测试参与者测试了护理计划原型。结果表明辅助医生制定个人计划的信息管理系统会因为网络基础设施性能的提高而变得高效，但是无法针对所有参与者即病人提出一个护理计划制定规范。

1.3.2 业务流程建模

另一个管理家庭护理计划的方法是业务流程管理，特别是业务流程建模符号（BPMN）作为护理计划管理过程和详细建模的工具并进一步地分析结果。这个想法也被其他研究人员所注意。其实最近提出一种叫做使用过程（Use Process）的新的方法。这种方法结合业务流程建模符号（BPMN）和统一建模语言（UML）中用例图结合的方法来满足建模需求。研究考虑到这些作家的地位，由于企业的成功运作是基于他们的业务流程，所以他们说明以业务流程为导向的方法来收集需求可能会导致在需求定义中客户参与度提高。他们开发出用户工程（UP）的方法来收集需求，其中包括基于符号元素的业务流程建模符号（BPMN）2.0 和统一建模语言（UML）1.5 的用例图的使用过程图（UPD）。

此外，他们提出了一套模板去描述使用过程图的元素。使用过程图的目的是呈现一系列功能来表现业务流程的活动。

Leonardo 项目组提出这种方法是基于以下几个阶段：首先，他们建模在 ASIS 模型中整个护理计划管理过程[4]。ASIS 模型见图 1-4。然后他们分析软件包 InformaCare 通过识别用户配置文件和功能来支持护理管理过程。之后他们分析护理管理过程中确定的关键绩效指标。最后他们进行了针对性的访谈和问卷调查来追查相关改进工程（模型）来引出一个新的业务流程模型，包括新工具的需求和定义性性能指标来控制整个系统[4]。

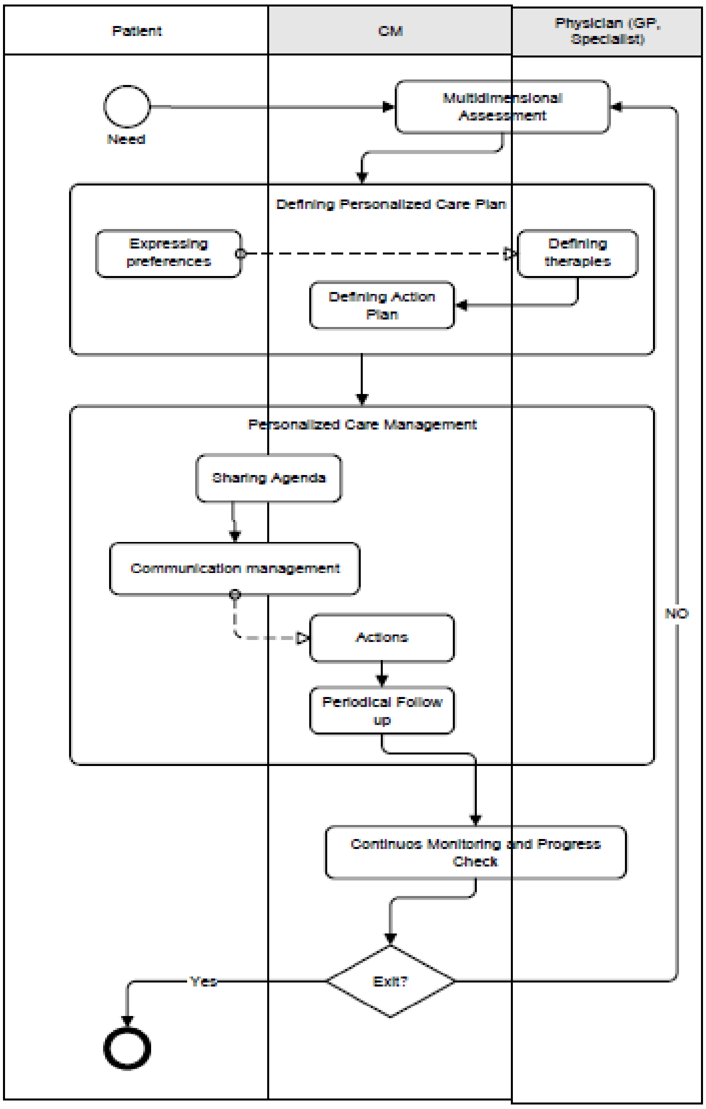

Patient
CM
Physician (GP, Specialist)
Need
Multidimensional Assessment
Defining Personalized Care Plan
Expressing preferences
Defining therapies
Defining Action Plan
Personalized Care Management
Sharing Agenda
Communication management
Actions
Periodical Follow up
NO
Continuos Monitoring and Progress Check
Yes
Exit?

图 1-4 类业务流程建模来管理护理计划[4]

一些研究表明，业务过程建模证明从商业的角度分析过程执行的难度。首先，从商业的角度来看虽然过程建模通过工作流自动实现，但是规范分析需要从实现的角度来呈现。虽然有一些工具和技术来分析业务流程(业务活动监督)，但是这些解决方法用他们自己语言来分析在施工系统中的问题（如 BPEL 和 Java）. 然后，当前业务过程分析解决方案通常只能性能分析（例如，时间运行，当前状态），但不是内部定义的基本活动（数据）。然而，当一个关键属性值被分配以采取互补性决定（例如，允许数据修改），整个业务过程建模将被否定。

所以管理家庭护理计划和传统的业务流程建模技术难以题，原因如下：

（1）家庭护理计划可以被看做是重复的活动（例如，医疗行为），这是一个在一定时间内重复的结合。然而，这些活动的制定根据不规则的时间表。活动的不规则性有两个原因。首先，活动通常根据不规则的需求演变。这种不规则的类型特变是通过加强或减弱的节奏实现的。然后，这种符合人类生活周期的活动完全符合人类生活周期的特点，特别是休息时间如周末，假期。这周类型的凹凸引起节奏的终端。所以不规则的活动规范需要合适的时间限制。

（2）家庭护理计划基本上都是非结构化的过程，在这个意义上，每个病人都有自己特定的护理计划。事实上，为每一个病人制定护理计划是完全针对个人的基础上，因为无论是从其病理或需求上来讲，每个病人是都是唯一的。因此，提前设计一套唯一的过程来描述所有病人的家庭护理计划简直是不可能的。换句话说，传统的方法模式中“建模一次，多次执行”针对家庭护理计划是不可行的。

1.3.3 Vaadin 框架

对于我们的项目，我们选择了很多形式为家庭护理计划制定者来构建智能工具, 比如网页框架（PHP，JSP），桌面程序或者手机程序。最后本着简化用户操作和开发者工作还有迎合制定家庭护理计划的需求，我们选择一种集合网页框架和桌面程序的方式—Vaadin 框架。Vaadin 是一个基于 Ajax 的网页应用开发框架。它通过 java 编程可以使开发人员打造高品质的用户界面，无论是在服务器端还是在客户端。它提供了一套用户界面组件相关的库和一个干净的框架用于创建自己的组件库。重点是 Vaadin 框架的易用性、可重用性和可扩展性完全符合应用的要求。Vaadin 框架的运行架构如图 1-5。我们将使用 Vaadin 的框架，设计图形用户接口[12]。

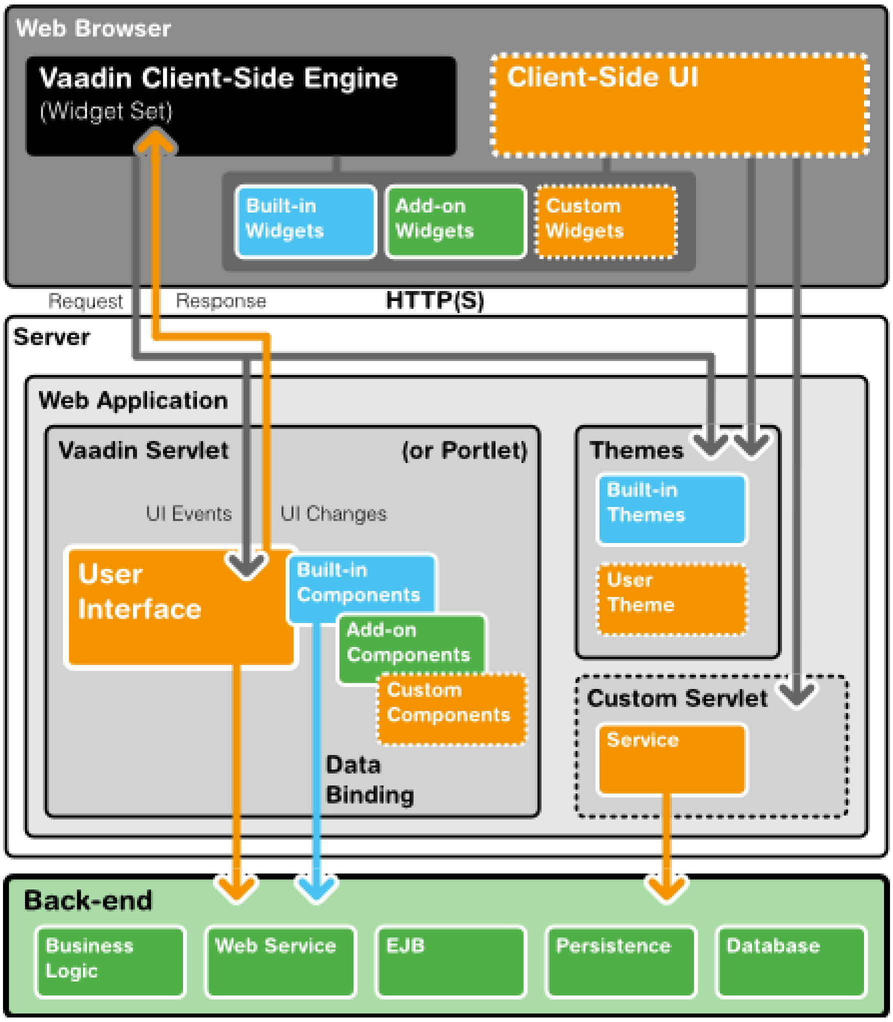


图 1-5 Vaadin 框架的运行架构

Vaadin 框架的主要优点是你可以管理整个界面通过 java 代码。这使得开发人员不需要学习其他复杂的框架例如 Strusts 或者 Java Server Faces 因为更多开发人员熟悉用 java 来开发项目。同时，Vaadin 的开发团队创建了大量的核心 UI 组件。开发人员只要通过简单的操作就可以生产自己想要的模型。还有，使用 Vaadin 框架不需要 CSS 或者 HTML5 来设置界面，因为所有的界面设计都可以自动生产并且 Vaadin 框架提供了大量主题取代 JavaScript 使界面更加美化。

1.3.4 时间自动机

为了检验家庭护理计划的合理性，我们引用时间自动机来检查我们的模型。时间自动机在 90 年代被拉杰夫阿鲁尔和大卫迪尔推出。这种形式主义是有各种类型时钟的有状态自动机的扩展。事实上，一个时间自动机被定义成一组节点

（状态），事务，操作及时钟的集合。节点表示该系统的状态，事务由事件符号标记，而时钟是零或者正数的变量的真实值。

定义 1:（时间自动机）时间自动机是一个七元组 $A=(S,s_0,\sum,X,Inv,T,F)$ 其中：S 是有限状态集合； $s_0 \subseteq S$ 表示初始状态； $\sum$ 表示动作的集合； X 是时钟变量的集合； $Inv:S \rightarrow \phi(X)$ 将一个状态映射为一个保卫公式，称为状态的不变量（invariant）； $T \subseteq S \times \sum \cup \{\varepsilon\} \times \phi(X) \times 2^X \times S$ 是有向边的集合，元素 (s,a,ϕ,λ,s') 便是状态 s 到状态 s' 的有向边;F 是数据变量的集合；

多个并发的时间自动机可构成一个时间自动机网络（timed automata network）. 网络中的多个时间自动机共享一些时钟变量和数据变量，但都有各自的状态. 格局（configuration）是用于描述网络中所有自动机运行时的状态（即全局状态）的概念。

人们研发出多种模型检测工具，如 UPPAAL、KRONOS[42]等. 这些工具为时间自动机的行为模拟、性质自动检测提供了强有力的支持。UPPAAL 是一个工具盒用来验证（通过图形仿真）和检查（自动模型检查）实时系统。模型检查工具 UPPAAL 是基于定时自动机理论并且他的建模语言提供额外的功能如有界整数变量和紧迫性。它包括两个主要部分：一个图形用户界面（在用户在上执行）和一个模型检查引擎。UPPAAL 的查询语言是一个定时计算逻辑树的子集合来用于指定属性来进行检查。在 UPPAAL 中，系统运用定时自动机来建模。所有时钟以同样速度记时来表示时间进度。时钟的实际值可以被测试或者复位。时间自动机模型见图 1-6。

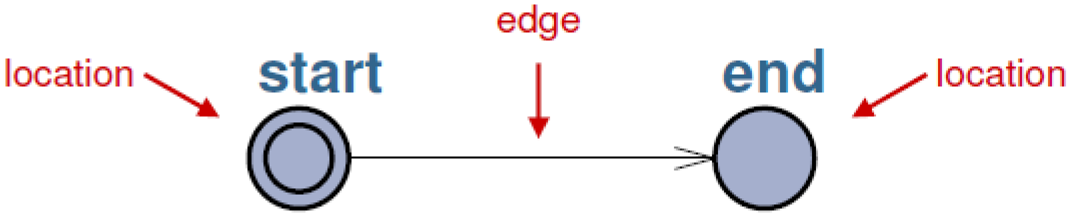


图 1-6 一个 UPPAAL 模型的例子

1.4 论文的主要内容

本文提出了一种新型的基于领域特定语言的方法来管理家庭护理计划，并开发了一个可视化和智能化的工具能简单、快速、准确地表达，监督和管理家

庭护理计划的制定过程。同时，利用模型检查工具 UPPAAL 来验证护理计划并提供有用的反馈给家庭护理计划的制定者。

本文将通过四个部分呈现。第一章给出家庭护理计划的概述和本项目的实际意义。这章还涉及介绍相关领域的工作，这将指导和提示笔者提出他的解决方案。第二章介绍了整个项目的总体需求，这些需求来源于和那些家庭护理计划的制定者面对面交流。第三章阐述了整个用来管理家庭护理计划的方法的设计，并详细解释了所采取的方法来提供了一个灵活的目标集中的解决方案来满足最终用户的需求。第四章介绍了如何一步一步地实施项目来解决实际工作并进行了测试和评估。

1.5 小结

本章主要产生了本项目的背景，来源和目的。然后，通过研究相同领域中相关的学术成功和理论来指导和启示本项目中新的方法的提出。最后，整片论文的主要内容和组织结构被详细说明。

第 2 章 系统需求分析

2.1 系统目标

本文的目标是提出一个新的方法来管理家庭护理计划并开发一个可视化和智能化的工具。该工具能够分析，检测家庭护理计划并最终给界面反馈。同时作为 Plas'O'Soins 的一部分，我计划为其他开发小组设计良好的查询接口和完整的数据库。来自家庭护理计划协调者的需求将完全被实现。如果需求正确的话，新的方法有责任找到一个解决方案。否则，这种方法将通过检查家庭护理计划来优化设计。这篇文章的难度在于设计一个有效的方法来解决来有效管理家庭护理计划这个问题。

2.2 系统功能需求

新方法的整体结构如图 2-1 所示。这种新提出的方法基于以下流程：首先，我们以统一建模语言中的类图对整个家庭护理计划进行建模,并根据需求的变化，反复修改元模型。然后，我们分析类之间的关系，并自动把它们翻译成类的对象。之后，我们通过对象关系映射建立对象与数据库的关系，其中数据库由关系数据库和知识库组成。关系数据库存储与家庭护理计划相关的数据，而知识库定义了一些默认的属性来实现系统智能化。在应用层面上，我们将设计一个富联网应用的工具（平台）来简化家庭护理计划制定人员工作并收集病人和其家庭护理计划的信息。工具中的每一个组件将和每一个类对象相对应。最终通过一些家庭护理计划的时间约束，家庭护理计划将跟踪新的业务流程相关联的模型改进的算法产生时间自动机网络。这样既可以重新定义新工具的要求又可以定义性能指标来控制整系统。

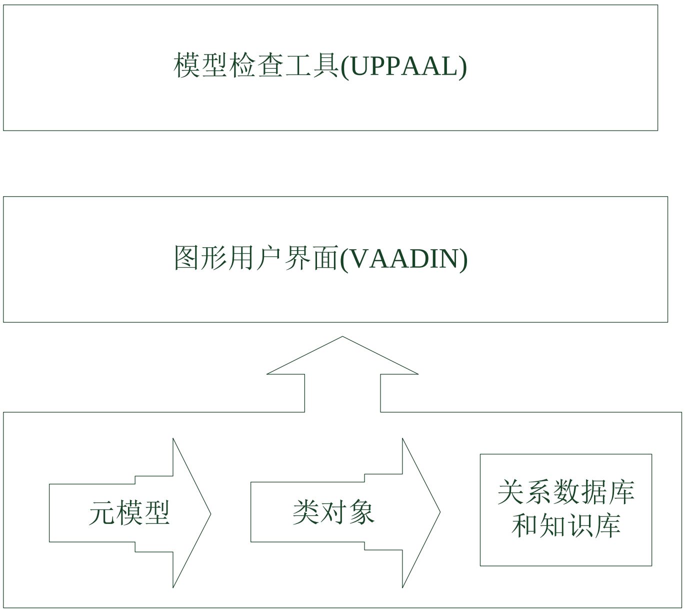


图 2-1 新方法的整体架构

2.2.1 从元模型到关系型数据库的转化机制

大部分的软件开发商和工业集团提供很大的努力去结合面向对象技术和关系模型已满足软件开发过程的需求[10][11]。从元模型到关系型数据库的转化目的在于把抽象模型转换成可以操控数据的关系表。一共分为两个过程：从元模型转换成类对象和建立类对象和关系型数据库的关系。元模型包括构成面向对象的基础概念,如元类、元属性、元操作、关联等来定义在 UML 图形中所用的全部语法单元以及它们的构成关系,即 UML 图形的语法结构。元模型详细地描述了项目的需求模型和实体之间的关系。类图是元模型的实例化。我们建立完元模型后采用类图的方式来呈现。所以以后当需求进行变更，只需要更改元模型。

具体需求如下：

- 用 Rational Rose 或其他统一建模语言设计工具去建立模型并根据变化的需求去修改它。
- 以 java 代码的形式自生成类并检查每个类图的语义。
- 通过给 java 代码加注释或者编写匹配文件的方式来产生关系型数据库。
- 对于知识库，需要手动导入需要的数据。
- 通过对象关系映射的方法建立类与数据库的关系。
- 提供一些接口给图形用户界面，以方便图形用户界面从数据库读数据。

2.2.2 图形用户界面

用户界面应该是 BS 结构的网页应用程序。他用来描述整个家庭护理计划过程。图形用户界面的使用者主要是专业医疗协调员。这些协调员主要负责制定和全程监控制定过程并检查他们的信息是否准确。整个图形界面见图 2-2。作为家庭护理计划协调人员，他们查询一个现存病人的信息和护理计划或者为一个新的病人创建新的家庭护理计划。当护理计划制定完以后，可以根据需求对其进行分类。

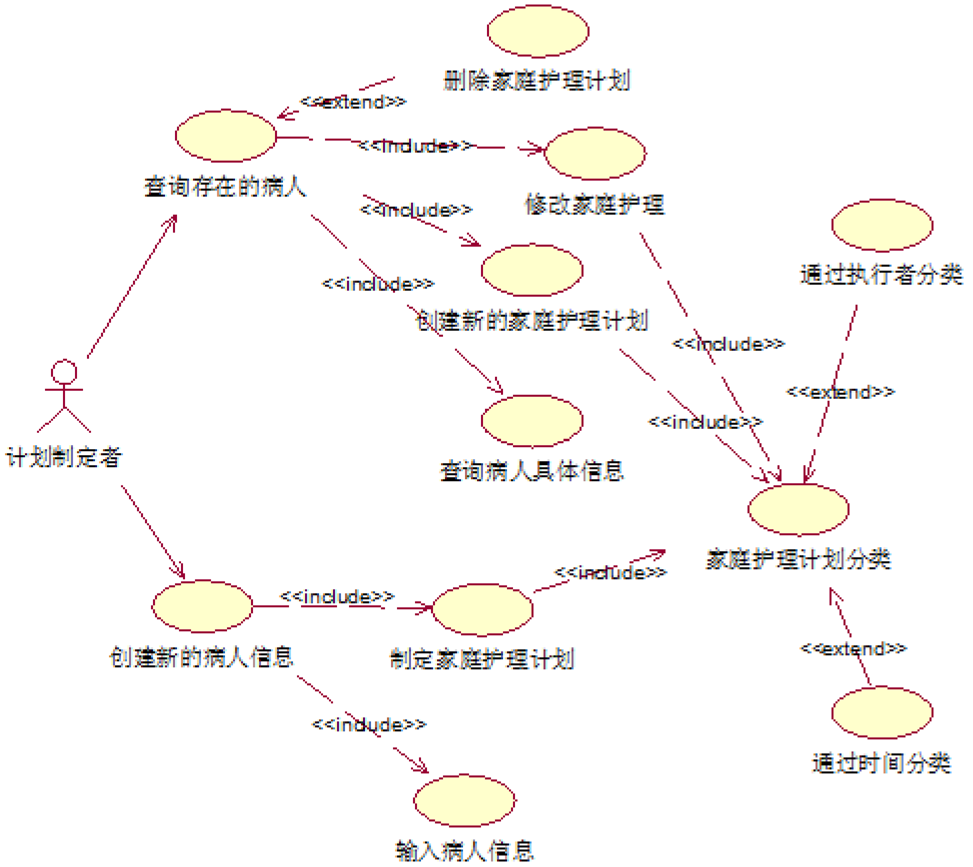


图 2-2 制定家庭护理计划用例图

对于病人信息，具体信息通过表 2-1 描述。病人的信息将被划分成个人信息，地址信息和制定信息三种。个人信息主要包括姓名，年龄等基本信息，其中提出了身体质量指数的概念来衡量个人的身体情况。地址信息主要是精确地描述本人的家庭住址，这将方便护理人员到家中进行护理。制定信息是指明填写病人信息的人员和信息以及制定护理计划的人员。

对于制定计划的过程，我们通过领域特定语言的方式来解决问，在家庭护理计划中主体是活动这一概念。每个活动都提供了和活动相关的描述，尤其是

时间的限制。这些限制描述了活动必须发生地时间。更准确地说，时间约束是一个三元组（时期，重复频率，时间范围）。其中，“时期”是制定活动发生的区间；“重复频率”是指在区间内某几天重复出现这项活动；“时间范围”是指在一天之中在哪些时间段该活动可能发生。

表 2-1 病人信息详细介绍

属性名	**衡量单位**	**案例**
姓名		张三
生日	日/月/年	01/03/1988
年龄	通过生日计算	25
性别	男/女	男
身高	米	1.72
体重	公斤	85
身体质量指数	$BMI = {weight}/{height^2}$	28.73
电话	区号+电话号码	0033064267777
地址	Street + city + post code + detailed building and apartment	
街道名和号码		幸福大街 33 号
邮编		33000
城市		克莱蒙克朗（法国）
建筑		人民大厦
房间号		房间 243
区域	知识库提供	克莱蒙费朗南部
修改人	默认（计划制定者）	Kahina
修改时间	默认（系统当前时间）	08/05/2013

我们用 BNF（巴克斯范式）来描述如下：

```
Period
<period>::=<starting-date>”-”{<ending-date>}?
<starting-date>::=”today”|<date>
<ending-date>::=”today”|<date>
Days
```

```
    <days>::={<date>+|{<day-of-week>}+|<specific-day>
    Where
    <date>::=/a date with the format dd/mm/yy/
    <day-of-week>::=”Monday”|”Tuesday” |”Wednesday” |”Thursday” |”Friday” |”Saturday”
|”Sunday”
    <specific-day>::=”Odd-day”|”Even-day”|”Holiday”
    Time ranges
    <time-ranges>::=(<Integer>”times”)|{<Interval>}+”nothing”
    where
    <integer>::=/number of occurrneces of the activity in the day/
    <interval>::=<string-form>|<pair-form>|<hour-form>
    <pair-form>::=”moring”|”midday”|”afternoon”|”evening”||”night”
   <hour-form>::=<hour>/an hour with the format hh/mm/
```

对一个活动来说，我们通过一个帮助病人“穿衣服”这个活动的例子给它详细的描述。详情请见表 Table 2-2. 所以整个家庭护理计划将使用一个活动表来进行描述。

表 2-2 “穿衣服”的时间域

时期	时间范围	重复频率
从 01/01/2013 到 03/31/2013	早晨	每天
		周一到周五
		周二在 8 点
	傍晚	一周除了周六
		假期

对于分配完的家庭护理计划，协调者可以根据护理人员类型，活动或者时间范围来进行分类。我们也同样使用一些表格来描述这些分类信息。可以说，这些表格拥有和活动表同样的信息但是不同样的结构而已。

除了活动这个概念，我们的领域描述性语言还有一个参与（intervention）这个概念。参与是指一个被安排在一起的活动的集合。在家庭护理方案中，它通常指被相同的人在相同的时间执行的活动集合。参与可以手动指定或者通过活动来自动计算生成。它主要用于计划协调员来验证计划的可行性。

2.2.3 模型监控和检查模块

当家庭护理计划制定完成之后，我们将使用模型检测工具来检测模型的正确性。首先，护理计划的基本活动通过适当的算法将被被转化成时间自动机。然后，一个动态的护理计划将被生成的足迹构成。每个护理计划中的活动都可以匹配成时间自动机。作为时间自动机的输入，一个基本活动的详细规格{activity name（活动名），period（时期），days（重复频率），time rangs（时间范围）}。

这个模块的具体需求：

1) 作为时间自动机的输入，{activity name（活动名），period（时期），days（重复频率），time rangs（时间范围）}将从数据库中获得数据。

2) 使用 UPPAAL 模型检查工具生成自动机网络来验证计划合理性。

3) 通过 UPPAAL 模型检查工具的分析结果，修改的意见将反馈给实施制定计划的人。

4) 设计和实现一个有效地算法来生成相应的时间自动机。

5) 根据基本的活动来自动生成整个护理计划，同样，家庭护理计划可以通过时间自动机表述。这将通过使用活动自动机的异步产品。

2.3 系统非功能需求

非功能需求如下：

- **可靠性**: 该方法应确保家庭护理计划制定的可行性和简化协调员的操作，提高其工作效率。

- **自动化性**: 该方法应在护理计划的管理过程中完全实现自动化即从现存的模型开始完全自动化生成有效地护理计划。

- **实时性**: 该方法应尽量减少产生家庭护理计划的使用时间并为使用者提供反馈。

- **可扩展性**: 一旦需求被改变，也就是，新的元模型或者知识库中的新信息被添加或者原有的需求被删除，整个过程将不会被改变。

- **代码可读性**: 因为本项目是一个子项目，所以要求在软件编码过程中，所有的类、方法以及关键的算法都比须添加详细的说明。软件的源代码必须保证其可读性及可理解性以保证软件的后续开发者能进行代码阅读。

- **智能化：** 这个需求是指当基层医疗计划协调员输入信息时，系统会提供智能联系功能。因为很多信息保存在知识库中，所以用户可以根据提示进行输入

信息，制定计划。

2.4 小结

本章对项目的需求进行了阐述及分析。

本项目的需求根据需求类型可分为功能性需求及非功能性需求。在功能性需求中又通过模型转化机制需求，用户界面需求和模型检查需求来进行阐述。

第 3 章 系统设计

根据需求分析，本项目的设计阶段可分为三个部分。首先，所有的类图形式的元模型将根据需求进行修改并转化成数据库和知识库。其中，模型转换法应当首先确立来建立准确的数据库。其次，我们将设计一个图形用户界面来监督家庭护理计划的制定过程。在设计界面过程中，使用了改进过的 V 模型过程管理技术因为需要和用户保存联系。在最后一部分中，使用数据库中关于家庭护理计划的数据，将自动生成时间自动机来分析和监控护理计划的制定并智能地修改用户界面。

3.1 家庭护理计划的领域特定语言

领域特定语言（DSL）是一种在一个特定的领域中解决问题的语言。我们提出的 DSL 通过护理协调员可来设计对于一个特定的病人家庭护理计划提供高层次的抽象。

在家庭护理计划中主要是两个重要的概念：活动和参与。活动是在病人家中提供的医疗服务。我们提出的 DSL 包括一系列通过我们的领域分析得到的预先定义的活动。参与是指一个被安排在一起的活动的集合。在家庭护理方案中，它通常指被相同的人在相同的时间执行的活动集合。参与可以手动指定或者通过活动来自动计算生成。它主要用于计划协调员来验证计划的可行性。除此之外，DSL 还定义了专业领域的附加约束（例如，医疗知识本体等）。

3.2 系统架构的设计

图 3-1 描述了整个方法的系统体系架构图。整个体系被分为 3 个层次：数据库层，图形用户界面层和逻辑层。数据库层主要是针对用户的需求，修改元模型并将通过对象关系映射技术其转化成数据库和知识库。图形用户界面层主要是构建一个图形的描述性的过程来制定家庭护理计划，并将所得到信息保存到数据库层中。而在逻辑层中，通过数据库提供的信息，尤其是一个时间的三元组（period（时期），days（重复频率），time rangs（时间范围））。再通过模型检测工具 UPPAAL 生成时间自动机来检查计划的可行性并通过算法进行优化，最后反馈到制定计划过程中。

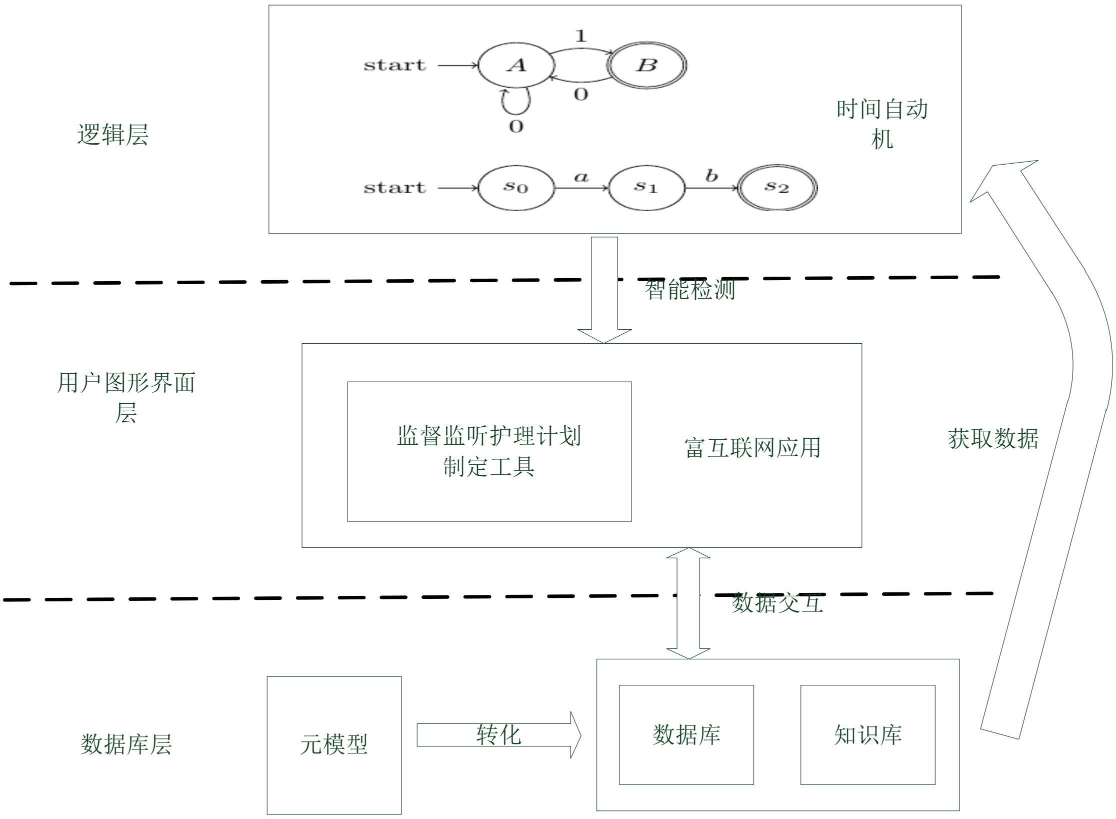


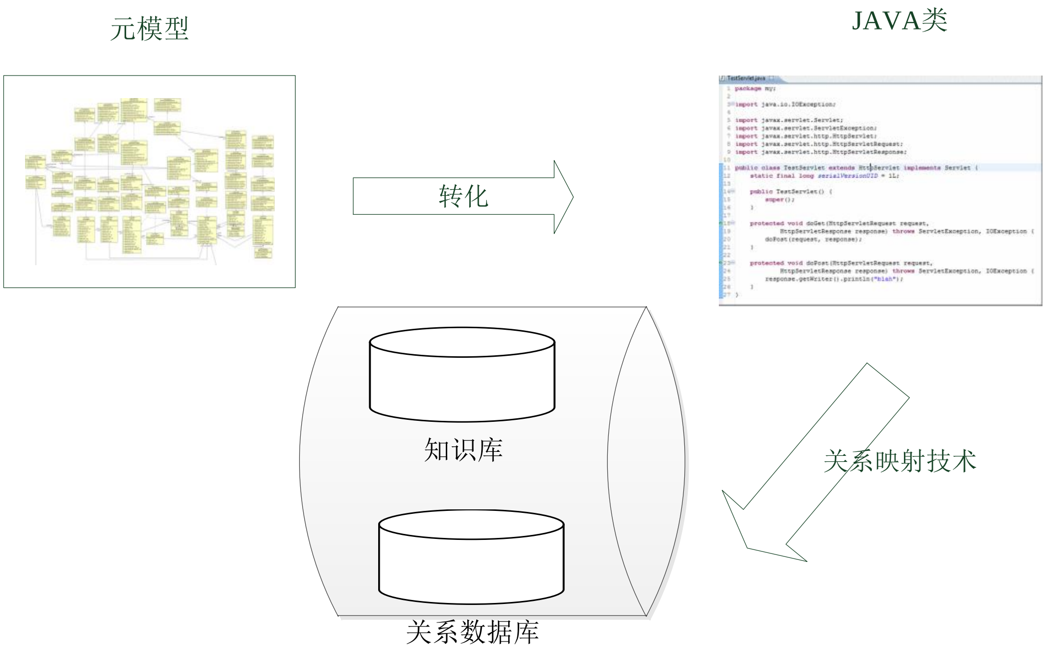
图 3-1 系统体系架构图

3.3 数据库层的设计

在数据库层中，整个过程体现如图 3-2. 首先，元模型应该被修订通过类图展示出来并转化成 JAVA 类，使用的工具是 Rational Rose。之后通过对象-关系映射技术，自动生成数据库并建立 JAVA 类和数据库联系。虽然我们使用的现有的技术，但是通过技术整合并且采用新的方式把元模型转化成了相应的关系型数据库。目的是无论以后如何需求变更，只需要更改元模型即可。增加了项目的可扩张性。

3.3.1 元模型的设计

正如需求分析中指出，整个家庭护理的流程是庞大的并不易进行指定的。所以本节提出了一个精简的与模型来进行描述。本章将介绍最为核心的一个包 projected care plan。其他的包都是对这个包的详细描述。



```
package my;

import java.io.IOException;

import javax.servlet.Servlet;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class TestServlet extends HttpServlet implements Servlet {
    static final long serialVersionUID = 1L;

    public TestServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        response.getWriter().println("blah");
    }
}
```

图 3-2 数据库流程设计

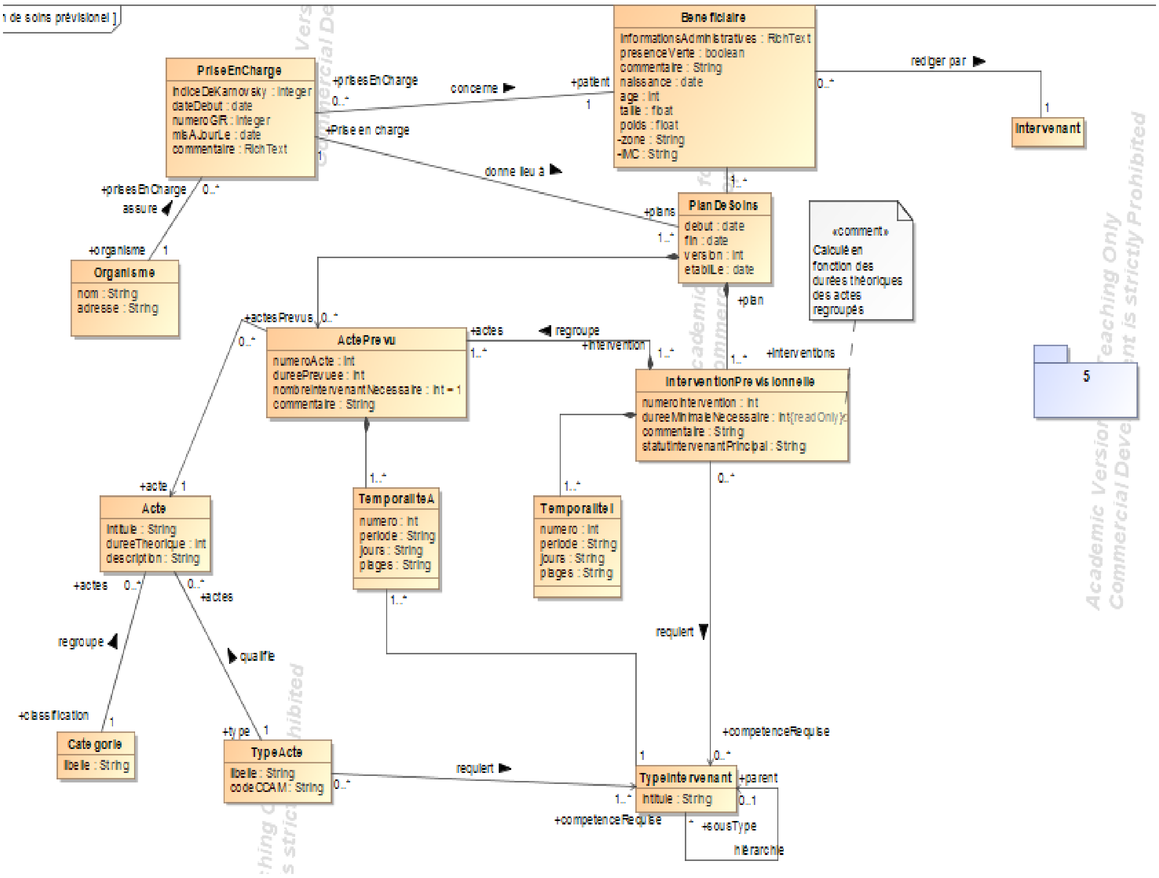


图 3-3 projected care plan 的包中类图

在整个预计护理计划包中，Beneficiaire (病人)类存储的是病人所有的情况，它继承于 Personne 类。里面的每一项就对应病人信息上的每个属性。每个病人对应着多个 PlanDeSoins(护理计划)类。PlanDeSoins 类中记录了护理计划的起始日期，终止日期，修改日期和版本号，其中版本号是为了每次显示的时候查找最新的版本号。整个护理计划是通过 ActePrevu（预计活动）来描述的。ActePrevu 类最核心的部分是保存在 Acte（活动）类中的活动名，类型和 TemporaliteA（时期）类中的三元组 period（时期），days（重复频率），time rangs（时间范围）。我们可以通过活动表产生干预表 Intervention(参与)类。参与表是对活动重新分类融合。PriseEnChange（支持）表中主要记录者如医院组织名称，执行人员或病人的相关信息。

建立好模型之后，我们通过 UML 建模工具 Rational Rose 建立类图并通过配置文件来实现自动化生产 java 代码，具体操作将陈述在该模块的实现部分。

3.3.2 JAVA 类和数据库的转换机制

通过对象-关系映射技术，由类图转化来的 java 类建立和数据库的联系。我们选择 Hibernate 框架作为我们对象-关系映射。主要是因为 1），Hibernate 会自动生成 SQL 查询。 2）Hibernate 提供了数据查询和检索设施，并能显着缩短开发时间，否则花手工数据处理 SQL 和 JDBC。 3）使应用程序移植到所有的 SQL 数据库。

通过整个过程，一个合理的数据库被最终创建。这个数据库包括存储性数据和需要人工输入数据的知识库。数据库具体情况如下，一些主要的数据库表将被下文介绍。

Acte 表 (见**错误!未找到引用源。**)包含活动的所有信息，如姓名，类型，持续时间和描述。Acte 表将作为知识库的一部分为用户提供智能联想提示。

表 3-1 Acte 表

属性	键型	类型	描述
id	Primary	int(11)	活动标识
intitule		Varchar(255)	活动名称
dureeTheorique		int(11)	活动持续时间
description		Varchar(255)	活动描述
categorieid	Foreign	int(11)	Categorie 表的标识
typeacteid	Foreign	int(11)	TypeActe 表的标识

TypeActe 表(见**错误!未找到引用源。**) and Categorie 表(见**错误!未找到引用源。**3) 存储和活动的相关信息如活动类型，活动类型等。他们同样是知识库的一部分。

表 3-2 TypeActe **表**

属性	键型	类型	描述
typeacteid	Primary	int(11)	活动类型标识
libelle		Varchar(255)	活动内容
codeCCAM		int(11)	

Table 3-3 Categorie Table

属性	键型	类型	描述
id	Primary	int(11)	种类标识
libelle		Varchar(255)	种类内容

Beneficiaire（病人）表（见表 3-4）包含病人所有信息，如同姓名，年龄，生日，电话等。对于地址，我们用街道，城市，住房号来描述。并且对于衡量病人健康情况的身体质量指数，我们利用病人的身高和体重来进行计算。Beneficiaire 表中很多得信息继承于 Personne 表。并且对于每个病人来说，都会为其分配护理人员（Intervenant 表）。对于一个护理计划来说每个病人可能被分配很多不同的护理人员。

表 3-4 Beneficiaire table

属性	键型	类型	描述
id	Primary	int(11)	病人 ID
Information Adminidtratives		varchar(255)	病人信息总汇
presenceVerte		int(11)	病人编号
naissance		Datetime	生日日期
commentaire		varchar(255)	病人描述
age		int(11)	年龄
taille		double	身高
poids		double	体重

续表 3-4 Beneficiaire table

属性	键型	类型	描述
rue		varchar(255)	街道
ville		varchar(255)	城市
zone		varchar(255)	区域
IMC		varchar(255)	身体质量指数
batiment		varchar(255)	病人公寓
appartement		varchar(255)	病人房间
nom		varchar(255)	病人姓名
telephone		varchar(255)	电话号码
mail		varchar(255)	邮件地址
intervenantid	Foreign	int(11)	护理人员 ID

PlanDeSoins（病人）表(见**错误!未找到引用源。**5) 包括关于家庭护理计划的信息如起始时间，结束时间等等。

表 3-5 PlanDeSoins 表

属性	键型	类型	描述
id	Primary	int(11)	护理计划 ID
debut		Datetime	起始日期
fin		Datetime	结束日期
version		int(11)	版本号
estabfiLe		Datetime	修改日期
priseenchargeid	Foreign	int(11)	PriseEnCharge 表 ID
beneficiaireid	Foreign	int(11)	Beneficiaire 表 ID

Acteprevu（预计活动）表 (见**错误!未找到引用源。**6)包括活动的一些基本信息。每个护理计划包括很多个不重复的活动如喂药，注射，穿衣的护理活动。活动的编号是从 0 自增的序列。.

表 3-6 Acteprevu 表

属性	键型	类型	描述
id	Primary	int(11)	活动 ID
numeroActe		int(11)	活动数
dureePrevuee		int(11)	活动的预计持续时间
nombreintervenantNecessaire		int(11)	负责该病人护理工作的人员数
commentaire		varchar(255)	活动描述
acteid	Foreign	int(11)	Acte 表 ID
plandesoinsid	Foreign	int(11)	PlanDeSoins 表 ID

Temporalite（时间区域）表(见**错误!未找到引用源。**7)描述一个活动的时间限制。它将作为时间自动机模型的输入。每个活动将有很多不同的时间限制。

表 3-7 Temporalite 表

属性	键型	类型	描述
id	Primary	int(11)	时间区域 ID
numero		int(11)	时间区域编号
periode		varchar(255)	时期
jours		varchar(255)	重复频率
plages		varchar(255)	时间范围
typeintervenantid	Foreign	int(11)	TypeIntervenant 表 ID
acteprevuid	Foreign	int(11)	Acte 表 ID

3.4 图形用户界面层设计

用户界面层是用来图像化地描述整个家庭护理计划制定过程。因为它主要为法国当地医院使用，图形用户界面有英语和法语两种语言。用户图形界面操作流程见图 3-4。首先，用户即计划制定者将选择他们熟悉语言。之后用户开始选择查看病人。如果病人档案存在，用户可以直接查看其家庭护理计划，否则用户必须在创建病人档案之后再为其安排家庭护理计划。家庭护理计划其实可以用一张活动表进行描述因为家庭护理计划是有一组活动组成的。当家庭护

理计划制定完成后，计划协调者可以再次修改家庭护理计划。当然也可与生成更适合他们观看的参与表。

家庭护理计划制定过程是最为核心的任务流程。制定家庭护理计划其实是填写一张活动表。在活动表中的每一行就对应着一个活动。制定家庭护理计划的概要流程见图 3-5。整个过程一个分为四个步骤。（1）选择活动。系统会给出一些默认的活动名称，同时，用户也可以输入一些新的活动名称。（2）选择每个活动的时间区间。时间区间包括时期，重复频率，时间范围三个重要属性。用户通过点击进行选择。（3）对时间区间表进行操作。当配置完一个活动以后，系统会生成这个活动的时间区域表。用户可以根据需要通过拖拽对其进行排序或者删除其中一个时间区域。（4）生成活动表。当添加多个活动以后，可以生成活动表。活动表是由活动的时间区间表加上知识库提供默认数据所组成。

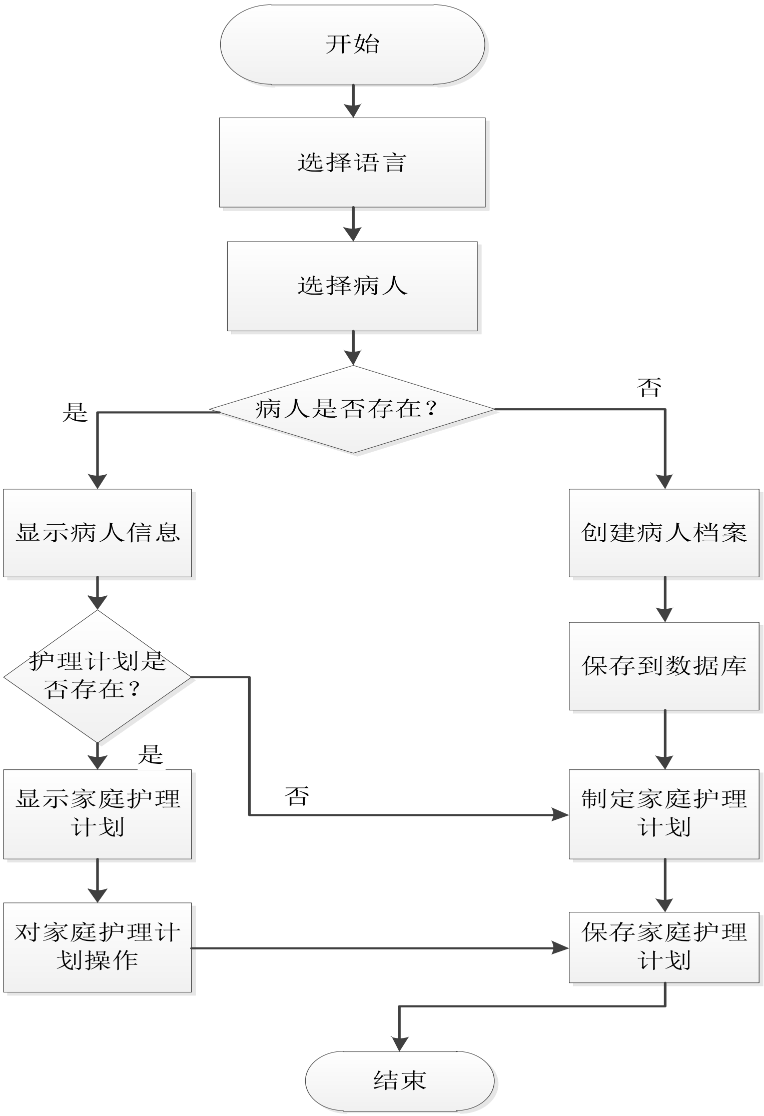


图 3-4 图形界面的操作流程

3.4.1 用户图形界面整体设计

用户图形界面主要是模型-视图-控制器（MVC）模型来进行设计用户整体框架。我们将所有的操作放在 Listener 包中作为控制器。对于视图，因为使用 Vaadin 框架，主界面都将写在 CarePlanManagerUI 类中并且利用 Vaadin 控件来生成其他界面一同放在 view 包中。因为用户图形界面的主题是由很多表组成的，所以我们设计了 TableModel 类来实时更新表格并且利用 Hibernate 来数据管理。

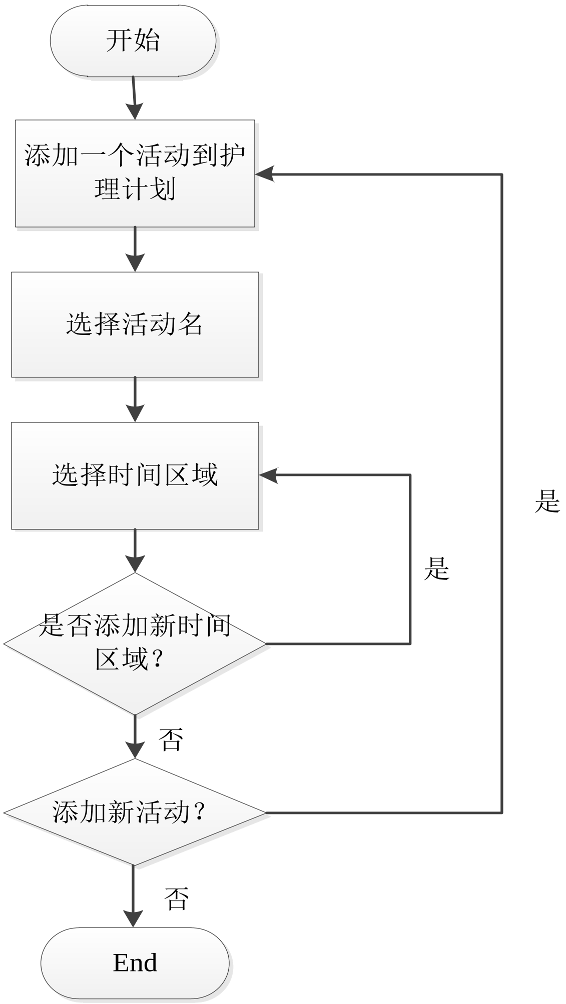


图 3-5 家庭护理计划概要流程图

整个用户界面流程时序图见图 3-6。家庭护理计划协调员最先想查看病人是否存在，首先，CarePlanManagerUI 对象调用 verify()方法。verify()方法将查询数据库中病人信息是否存在。然后 patientInformationManager 将调用 inform()方法发送病人的状态回 CarePlanManagerUI 对象。病人的状态一共有三种：状态为 0 表明病人不存在，在这种情况下 CarePlanManagerUI 对象将调用 record()

来说明创建病人档案；状态为 1 表明病人存在但是没有家庭护理计划，patientInformationManager 将调用 specify()方法新制定一个家庭管理计划；状态为 2 表明病人和家庭计划都存在，协调者就可以查看家庭护理计划了。在制定家庭护理计划的过程中，CarePlanManager 对象将调用 add()方法为活动添加时间区域。添加完之后系统调用 generate()方法长生参与表。并且参与表也可以反向生成活动表。

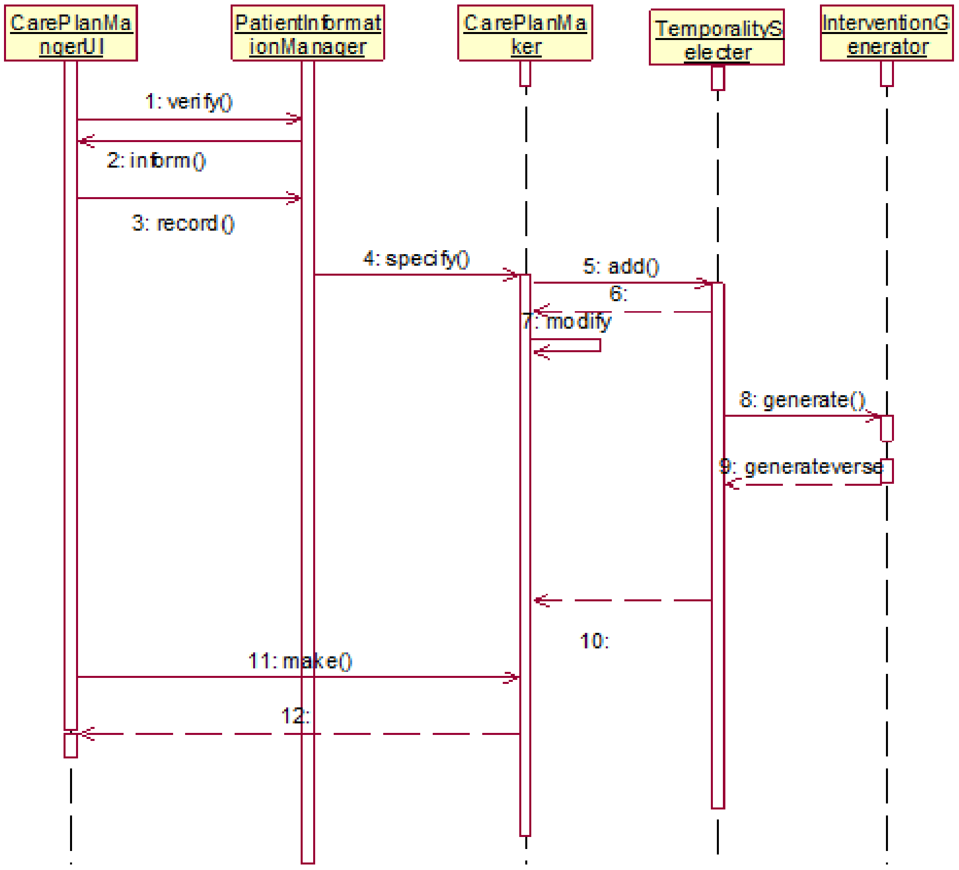


图 3-6 用户界面流程时序图

3.4.2 时间区间选择界面的设计

根据需求知道，整个家庭护理计划在界面层的具体体现是一个活动表。而活动表的核心就是为活动进行时间区间选择。因此时间区间选择界面是用户界面层的核心。并且我们将为活动选择时期，重复频率和时间范围。我们设定时期的格式是由(起始日期，结束日期)组成。日期的格式是按照法国时间格式“日/月/年”。对于重复频率我们定义为 4 种格式：详细日期（detailed time），每一天（Every day），一些天（Some day）和节假日（Festival）。其中针对假期，

我们统计了一年所有的法国假期，并把他们存在数据库中。而对于“一些天”是指重一周中选择一天或多天。如果用户选择联系的 3 天,结果将显示从第一天到最后一天如用户选择周一，周四，周五和周六，显示结果为“周一，周四到周六”。我们使用如下算法：

```
String[] str = the set of selected days
// switch days during weeks to the number {1-7};   {1,3,4,5,7}
Switch(str,strnumber);
sort ascending-> strnumber;
List List_single,List_start,;
Flag = 0;
For(i = 0; i<strnumber.length;i++)
{
     If(Strnumber[i]+1 = =strnumber[i+1]){
        Flag++;
     }
     else{
     flag =0;
     }
     If(flag>1)
     List_start<- flag;
     List_end<- i; }
     Else{
           Flag=i+1;
         List_single<-i;}
}
```

对于时间范围，一共两种选项：任意（nothing）和间隔（intervals）。“任意”是指一天的任意时间段都可以，而“间隔”包含 6 种情况：morning (7h00-11h00), noon (11h00-13h00), afternoon (13h00-18h00), evening (18h00-22h00), night (22h00-7h00 at tomorrow)和最为特殊的准确时间（detailed time）。用户可以选择其中一个或多个。但是一旦准确时间和其他情况发生冲突，系统将判断准确时间是否在其他间隔中并给出提示。所以在存储的时候，我们都以小时为单位进行存储。

3.4.3 参与表界面设计

参与表是根据护理人员的类型和时间范围对活动表的重新分类。我们设计了两个类型的参与表：详细参与表和简化参与表。在数据库中，所有的时间表示以字符串的形式存储，通过 switch()函数都将转换成以小时为单位的详细时间。之后家庭护理计划先通过时间范围进行分类之后再通过护理人员的类型来进行分类。这样将得到参与表的信息分类集合。最终我们通过这个过程可以得到活动与参与的对应关系。每个参与中包含着多个活动。所以参与的持续时间需要根据每个活动的参与时间进行计算。我们假设一个参与 I_1 包含 N 个活动 $(A_1 \ldots A_n)$。每个 A_i 的持续时间为 D_i 天。所以我们得到 I_1 的持续时间范围是 $(\max\{D_1,...,D_n\},\sum_1^n D_i)$。我们定义 Classifier 类归类家庭护理计划。最终参与表包含 1) 参与 ID，2) 时间范围， 3) 参与类型， 4) 护理人员状态， 5) 护理人员数量， 6) 护理人员名字，7) 参与的时间区间和 8) 参与的持续时间范围。

参与表界面生成过程时序图见图 3-7。InterventionListener 对象调用 switch()函数将数据库中数据转化成小时。得到结果后，analyze()函数进行自我比对找出相同的时间范围保存到不同的容器中。然后再通过 classify()根据不同的护理人员类型进行分类生成参与表。用户可以通过选择来展示完整参与表或者简易参与表。同时系统可以调用 genenrator()方法来反向生成活动表。

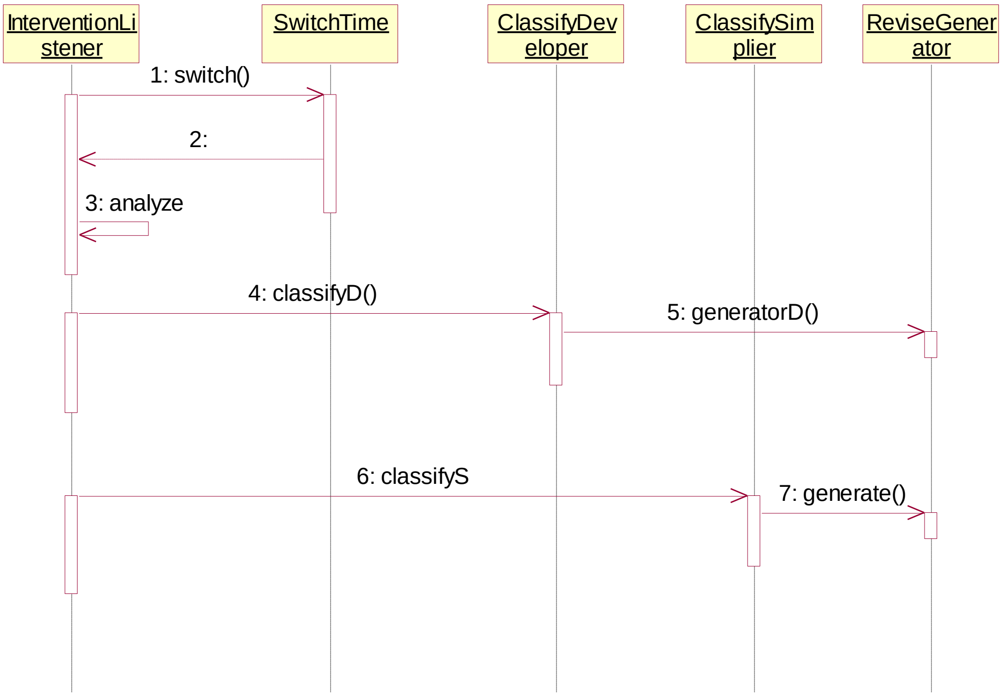


图 3-7 参与表界面生成过程时序图

3.4.4 用户图形界面

针对用户图形界面的开发，我们采取的是原型开发的方法,使用框架为Vaadin 框架。整个用户界面原型分为 3 个板块：用户信息板块，护理计划制定板块和计划时间轴板块。

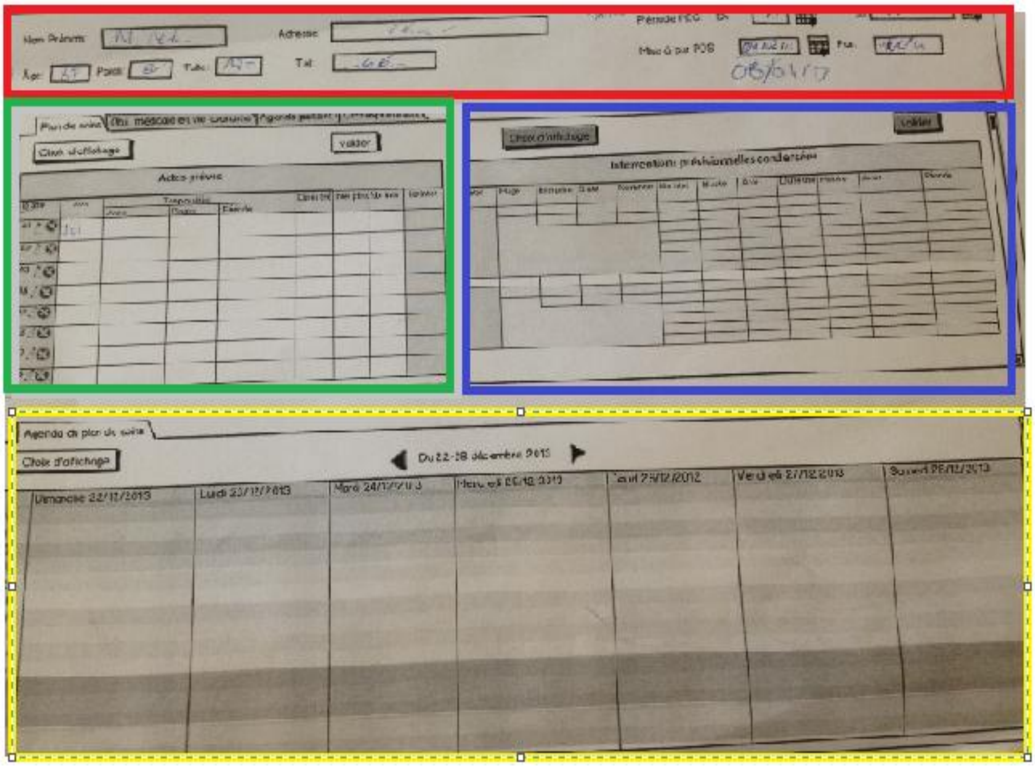
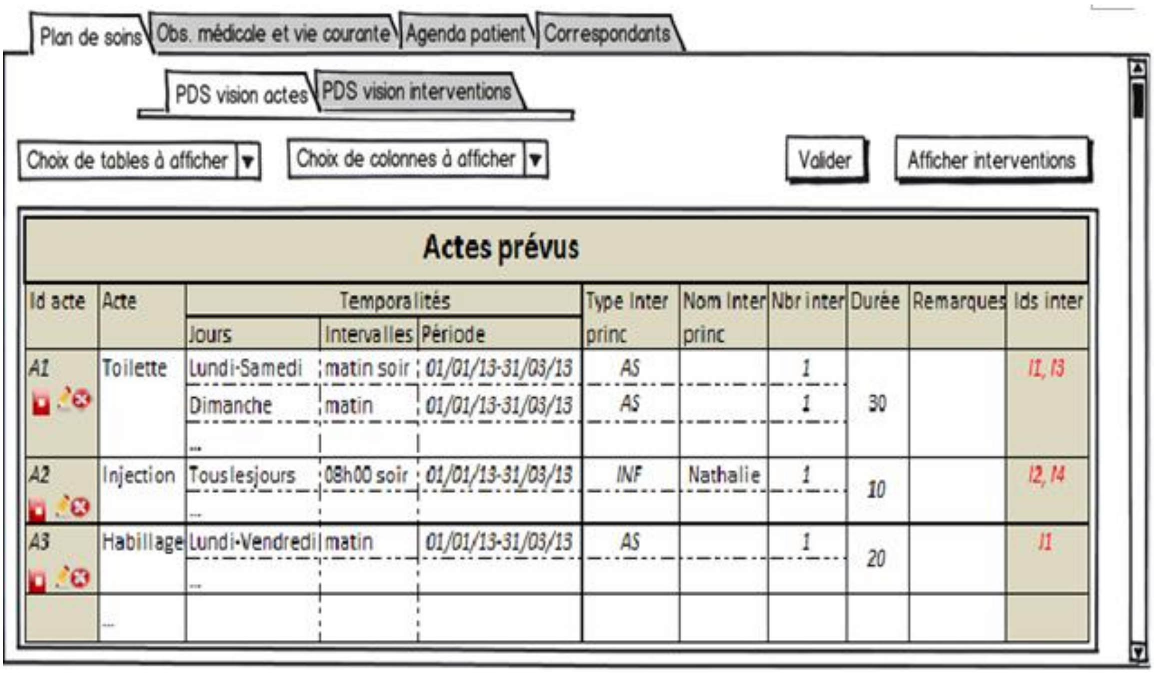


图 3-8 用户图形界面原型设计图

- 在红色的框中显示的是患者信息面板。用来记录病人的信息或者查询现存病人信息
- 护理计划制定模块一共有四个标签：护理计划，观察模式，病人日历和新闻速记。对于本项目主要关注第一个标签即填写活动表制定计划。护理计划及填写一个活动表。可以对活动表进行操作如选择表的一列，修改表的内容，删除一个活动等。活动表的原型图见图 3-9。
- 在护理计划中制定活动的时间区间是最重要的工作。我们设计了一个窗体来进行选择活动的时间区间（见图 3-10）。时间区间将根据时期->重复频率->时间范围的顺序进行选择。默认的时期是知识库提供并可以根据具体情况进行改变。对于重复频率用户可以选择 7 个格式：详细日期，每一天，一些天，奇数天，偶数天，每 N 天和节假日（Festival）。于时间范围，一共两种选项：

任意（nothing）和间隔（intervals）。用户可以通过点击复选框和单选框来进行选择。



Actes prévus

Id acte	Acte	Temporalités			Type Inter	Nom Inter	Nbr inter	Durée	Remarques	Ids inter
		Jours	Intervalles	Période	princ	princ				
A1	Toilette	Lundi-Samedi	matin soir	01/01/13-31/03/13	AS		1	30		I1, I3
		Dimanche	matin	01/01/13-31/03/13	AS		1			
		...								
A2	Injection	Touslesjours	08h00 soir	01/01/13-31/03/13	INF	Nathalie	1	10		I2, I4
		...								
A3	Habillage	Lundi-Vendredi	matin	01/01/13-31/03/13	AS		1	20		I1
		...								
	...									

图 3-9 护理计划制定模块原型设计图

3.5 逻辑层中模型检查模块的设计

在本节中，我们提出方法匹配家庭护理计划到时间自动机。我们提出了三步的方法，其原理如下：(i)我们需要对家庭护理计划进行监听和分析(ii)护理计划中基本活动将通过合适的算法转化时间自动机；(iii)通过 UPPAAL 进行模型检查。

分析和监督家庭护理计划。一旦护理计划用 DSL 构建，也就说，有了一个正式的模型描述护理计划，我可以通过护理计划的分析方面对其进行讨论。

➢ 护理计划的验证。针对某个特殊的护理计划来检测出可能的运行错误在理想的计划。例如，主要检查护理计划的可实现性，即检测在护理计划中活动符合计划中的限制并是否合理地安排和执行。

➢ 参与事件的验证。此分析主要考虑参与事件的时间约束和活动的限制条件的相互作用。例如，如果活动 A 出现在参与 I 中，那么我们需要检查 A 的时间约束是否和 I 一致。

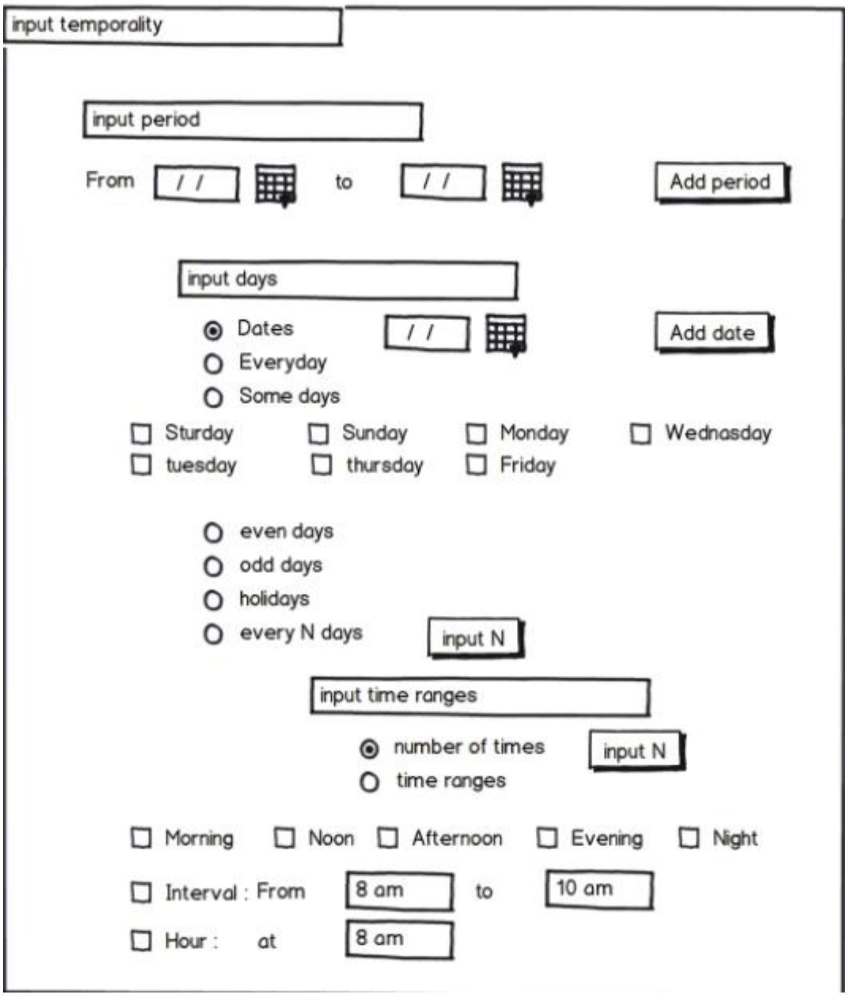


图 3-10 时间区间选择界面原型设计图

➢ 一致性验证。在某型情况下，一个规范必须满足活动间的依赖。我们必须彼此的一致性，如一个活动注射必须在过敏测试之后完成，一致性验证必须使用。

➢ 兼容性验证。主要是验证为病人设计的家庭护理方案中的时间范围是否病人或护理人员能够接受。

这节我们首先给出一些时间自动机的概念然后阐述如何匹配护理计划到时间自动机中的。

定义 1：（时间自动机）时间自动机是一个七元组 $A=(S,s_0,\Sigma,X,Inv,T,F)$ 其中：S 是有限状态集合；$s_0 \subseteq S$ 表示初始状态；Σ 表示动作的集合；X 是时钟变量的集合；$Inv:S \to \phi(X)$ 将一个状态映射为一个保卫公式，称为状态的不变

量（invariant）；$T \subseteq S \times \sum \cup \{\varepsilon\} \times \phi(X) \times 2^X \times S$ 是有向边的集合，元素 (s,a,ϕ,λ,s') 便是状态 s 到状态 s' 的有向边;F 是数据变量的集合；

每个活动都被映射到时间自动机 A_a 中。A_a 的构成基于活动 a 的规格并且特别是时间的限制。我们使用以下的形式来表述一个基本活动：{活动名称（Activity name），重复频率（days）,时间范围（time ranges）,时期（period）}。

有趣的是，护理计划也可通过时间自动机来描述。时间自动机就是用基本活动组成的。这种组合被获得通过时间自动机的异步产品。这种时间自动机的异步产物允许我们识别家庭计划的可能配置。下面我们介绍一下自动机异步产物。

定义 2：（时间自动机的异步产物）$A_1 = (S_1, s_0^1, \sum_1, X_1, Inv_1, T_1)$ 和 $A_2 = (S_2, s_0^2, \sum_2, X_2, Inv_2, T_2)$ 是两个时间自动机。A_1 和 A_2 的产物 $A_1 \times A_2$ 是时间自动机 $(S_1 \times S_2, s_0^1 \times s_0^2, \sum_1 \cup \sum_2, X_1 \cup X_2, Inv, T)$，其中 $Inv(S_1,S_2) = Inv(S_1) \wedge Inv(S_2)$ 并且 T={ $((s_1,s_2),a,\phi,\lambda,(s'_1,s'_2)) : ((s_1,a,\phi_1,\lambda_1,s'_1) \in T_1$ 和 $s_2 = s'_2$ or $(s_2,a,\phi_2,\lambda_2,s'_2) \in T_2$ and $s_1 = s'_1)\}$ 。

通过时间自动机我们来设计：

- 护理计划的验证。通过检查护理计划的可实现性来检查相应自动机的空虚问题。
- 参与事件的验证。参与事件被形式地描述成自动机。然后参与时间的检测是通过检查活动自动机和参与自动机的交点。
- 一致性验证。依赖性也将产生自动机并且一致性的检查是参与自动将和依赖自动机的组合。
- 监控验证。检查执行轨迹的契合程度可以减少时间自动机中语言语言的定义。

3.6 关键技术

本课题共使用了 3 种关键技术。

1) 元模型到关系型数据库自动转换方法。本项目利用两套独自的技术并整合到一起以适应课题需要。

2) DSL 方法构建用户图形界面技术。本项目利用提出了一种基于 DSL 方法来构建图形界面并采用 Vaadin 框架对其进行实现。

3) 模型检查技术。我们定义了时间自动机模型并使用三元组作为输入产生我们的模型。通过分析将计划匹配到时间自动机中并进行检查。

3.7 小结

本章对项目进行了设计。设计可以分为三个部分：对数据库层的设计，对图形界面层的设计和对逻辑层中模型检查的设计。

第 4 章 系统实现与测试

4.1 系统实现环境

CPU：Mobile Dual Core Intel Core i7-2637M, 2500 MHz (25 x 100)，

系统内存：3964 MB (DDR3 SDRAM)，

主板：Intel Direct Media Interface v2.0，

另外，软件平台环境如下：

操作系统：Microsoft 7 SP2,

开发平台：Eclipse Juno，

数据库：MySql，

编程语言：Java, domain-specific language，

模型检查工具：UPPAAL，

统一建模语言工具：Rational Rose IBM, MagicDraw，

富互联网应用程序框架：Vaadin框架，

4.2 数据库层实现

首先我们通过 Rational Rose 中的类图来绘制我们建立好的模型。之后通过配置 Rational Rose 中的设置来自动产生代码。

1) 将 java 类加入到模型中的 java 组件

2) 对类图进行语法检查

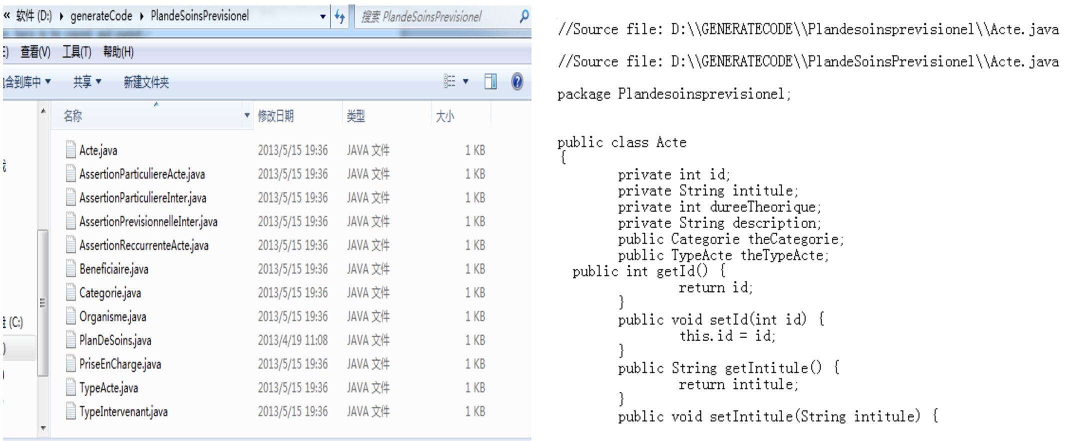
3) 设置 *CLASSPATH* 环境变量

4) 设置 *Tools-Java-Generate Java*.相应参数

5) 备份文件

6) 生成代码

图 4-1 展示了通过整个过程生成的结果。


« 软件 (D:) ▸ generateCode ▸ PlandeSoinsPrevisionel
搜索 PlandeSoinsPrevisionel
查看(V)
工具(T)
帮助(H)
包含到库中 ▾
共享 ▾
新建文件夹

名称	修改日期	类型	大小
Acte.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionParticuliereActe.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionParticuliereInter.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionPrevisionnelleInter.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionReccurrenteActe.java	2013/5/15 19:36	JAVA 文件	1 KB
Beneficiaire.java	2013/5/15 19:36	JAVA 文件	1 KB
Categorie.java	2013/5/15 19:36	JAVA 文件	1 KB
Organisme.java	2013/5/15 19:36	JAVA 文件	1 KB
PlanDeSoins.java	2013/4/19 11:08	JAVA 文件	1 KB
PriseEnCharge.java	2013/5/15 19:36	JAVA 文件	1 KB
TypeActe.java	2013/5/15 19:36	JAVA 文件	1 KB
TypeIntervenant.java	2013/5/15 19:36	JAVA 文件	1 KB

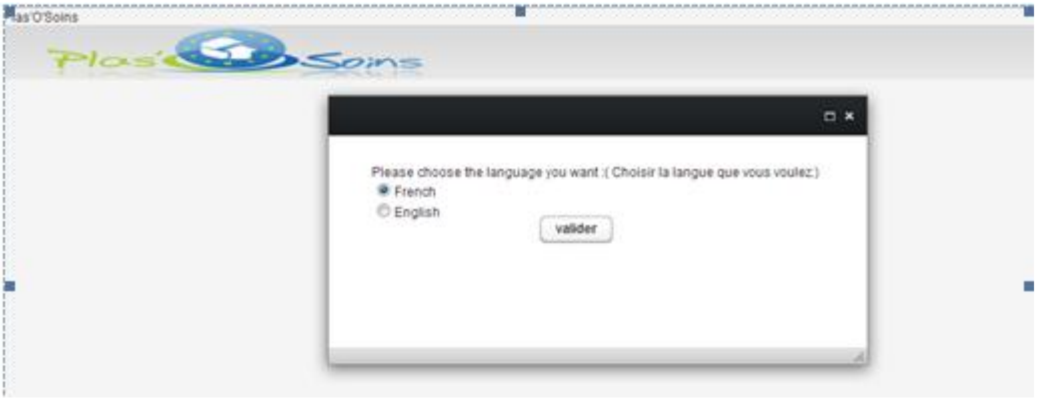
```java
//Source file: D:\\GENERATECODE\\Plandesoinsprevisionel\\Acte.java

//Source file: D:\\GENERATECODE\\PlandeSoinsPrevisionel\\Acte.java

package Plandesoinsprevisionel;


public class Acte
{
        private int id;
        private String intitule;
        private int dureeTheorique;
        private String description;
        public Categorie theCategorie;
        public TypeActe theTypeActe;
  public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
        public String getIntitule() {
                return intitule;
        }
        public void setIntitule(String intitule) {
```

图 4-1 由 PlandeSoins 包的类图自动生成的代码

之后我们通过 hibernate 框架建立了 java 类和关系型数据的联系。通过编写 hibernate 映射文件来反映类之间的关系，之后自动导出关系型数据库。

4.3 用户图形界面层的实现

当用户运行程序时，首先进行语言选择（见图 4-2）。因为是针对法国护理中心开发的项目，所以默认的语言是法语。为了实现语言转换功能我定义了 constant 其中含有 ConstantEnglish 类和 ConstantFrench 类来存储相应的信息。



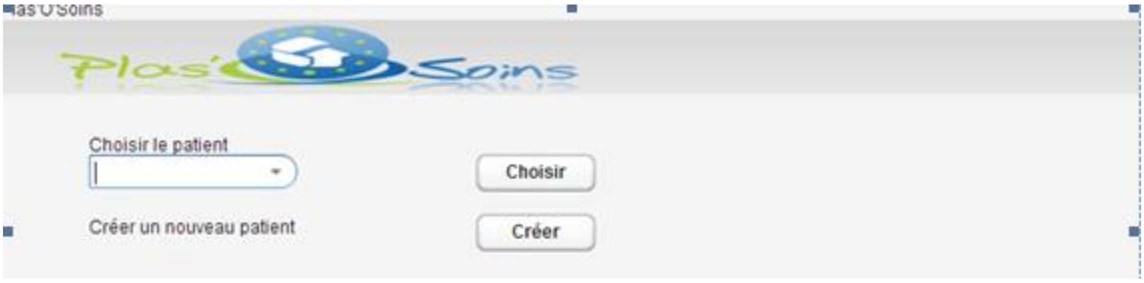
Please choose the language you want :(Choisir la langue que vous voulez:)

● French

○ English

图 4-2 语言选择窗体

当医疗计划协作者选择完语言，他可以选择查询现存病人或创建一个新的病人。现存病人列表从知识库中提供。

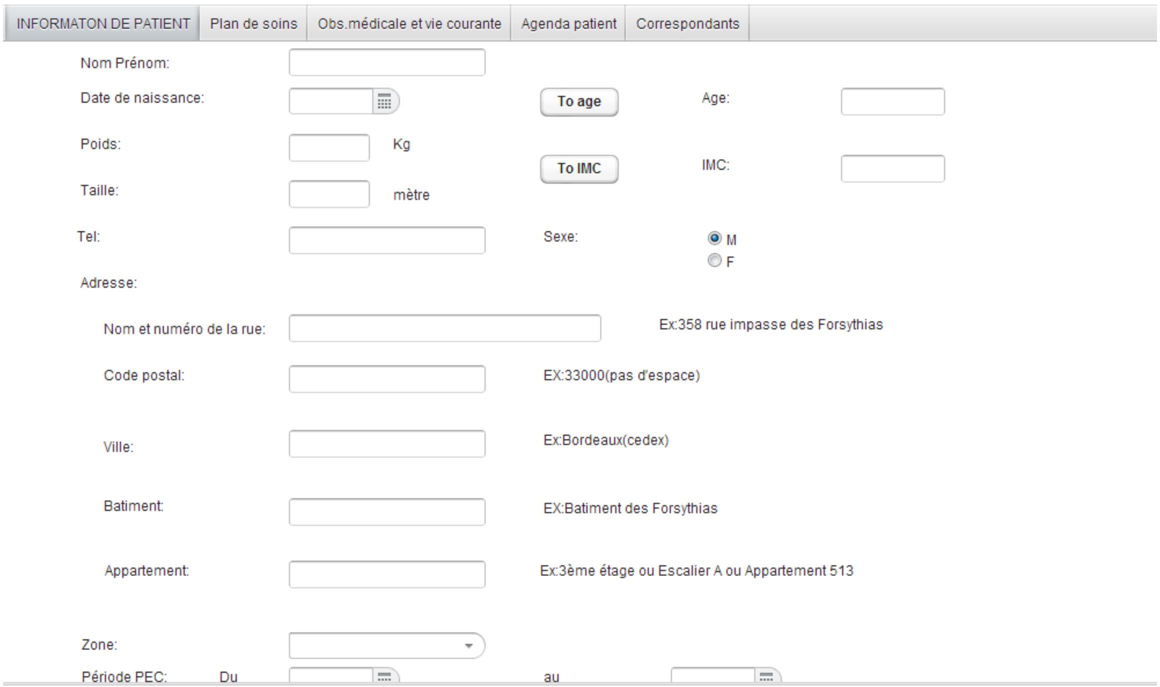


Choisir le patient

Créer un nouveau patient

图 4-3 病人选择窗体

用户信息窗口见图 4-4。首先用户输入病人的姓名，生日，身高，体重，性别，电话号码等基本信息。通过生日可以计算出年龄，而通过身高和体重，可以计算出病人的身体质量指数。因为家庭护理计划提供的是上门服务，因此病人的地址应该被详细描述：街道号和名，邮编，城市，住房信息。同时将他的地址用区域来归类。最后填写创建人姓名和日期。如果病人有个人信息在数据库中，那么用户信息窗体使用来展示信息的。否则用户信息穿衣是用来创建一个新病人的档案。



Nom Prénom:

Date de naissance:

Age:

Poids: Kg

IMC:

Taille: mètre

Tel:

Sexe: ● M ○ F

Adresse:

Nom et numéro de la rue: Ex:358 rue impasse des Forsythias

Code postal: EX:33000(pas d'espace)

Ville: Ex:Bordeaux(cedex)

Batiment: EX:Batiment des Forsythias

Appartement: Ex:3ème étage ou Escalier A ou Appartement 513

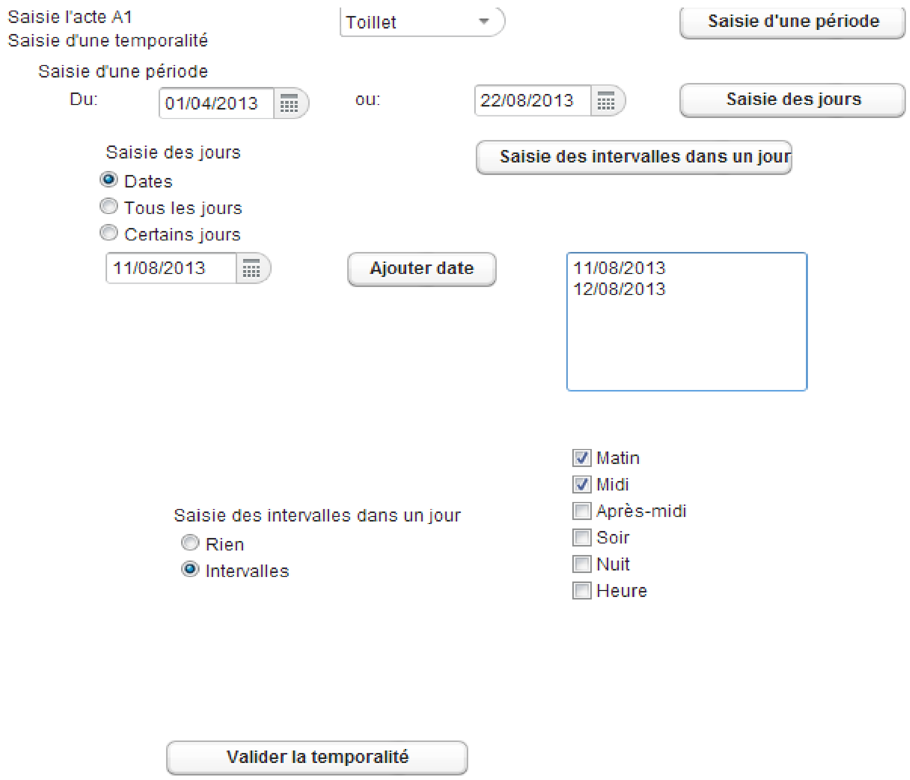
Zone:

Période PEC: Du au

图 4-4 病人信息窗口

家庭护理计划制定表示由活动表组成的。为了构建这个活动表，我们需要每次为其添加不同的活动。首先，点击添加活动按钮，添加活动窗口将展现在

你面前。用户选择活动名之后，开始为该活动添加不用的时间区间。选择完之后点击确认按钮。活动表的制定如图 4-5 至图 4-7。



Saisie l'acte A1 Toillet

Saisie d'une temporalité

Saisie d'une période

Du: 01/04/2013 ou: 22/08/2013

Saisie des jours

● Dates
○ Tous les jours
○ Certains jours

11/08/2013

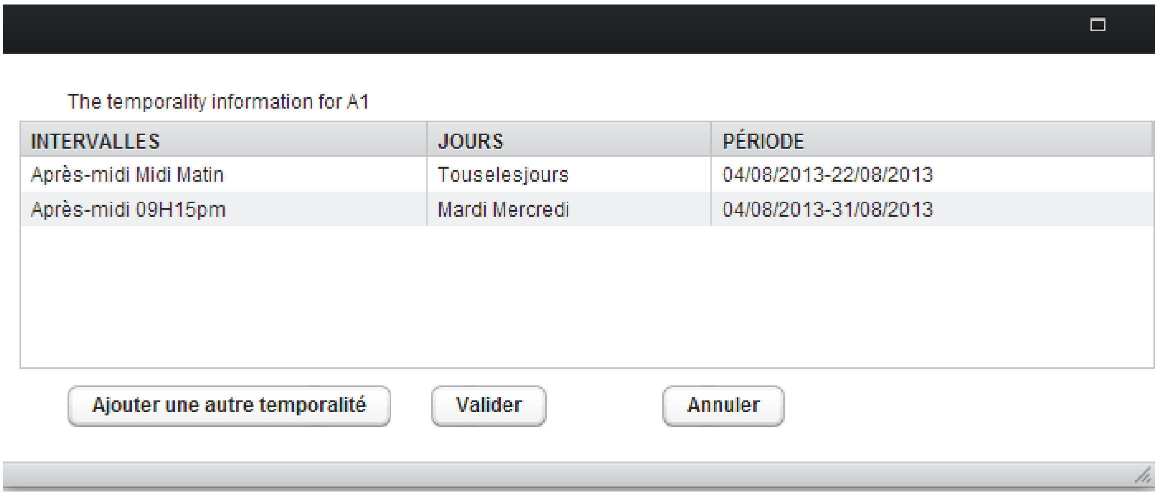
11/08/2013
12/08/2013

Saisie des intervalles dans un jour

○ Rien
● Intervalles

☑ Matin
☑ Midi
□ Après-midi
□ Soir
□ Nuit
□ Heure

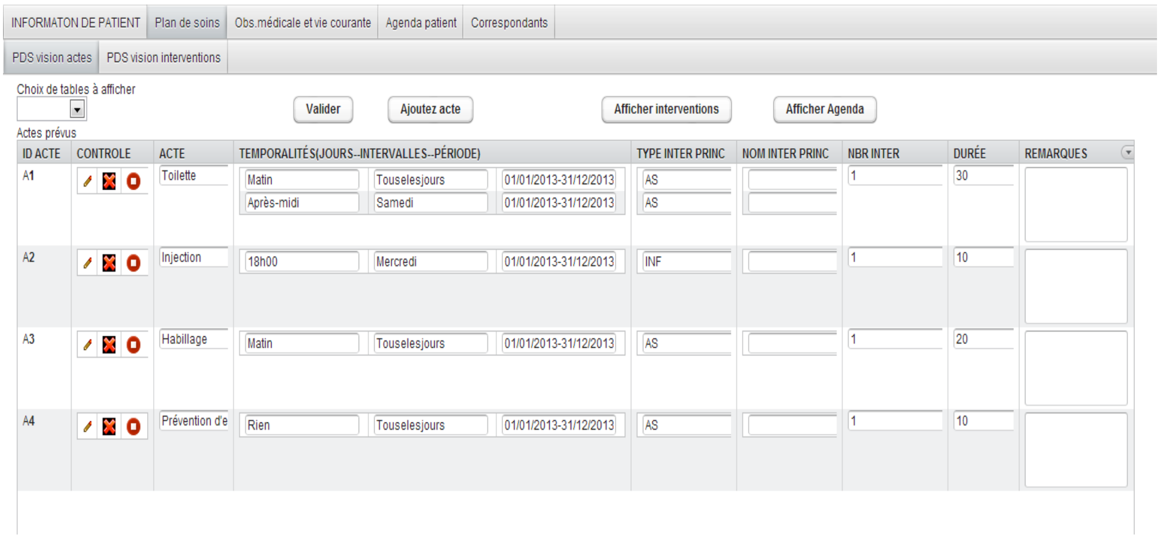
图 4-5 添加活动窗口



The temporality information for A1

INTERVALLES	JOURS	PÉRIODE
Après-midi Midi Matin	Touselesjours	04/08/2013-22/08/2013
Après-midi 09H15pm	Mardi Mercredi	04/08/2013-31/08/2013

图 4-6 时间区间选择展示窗体

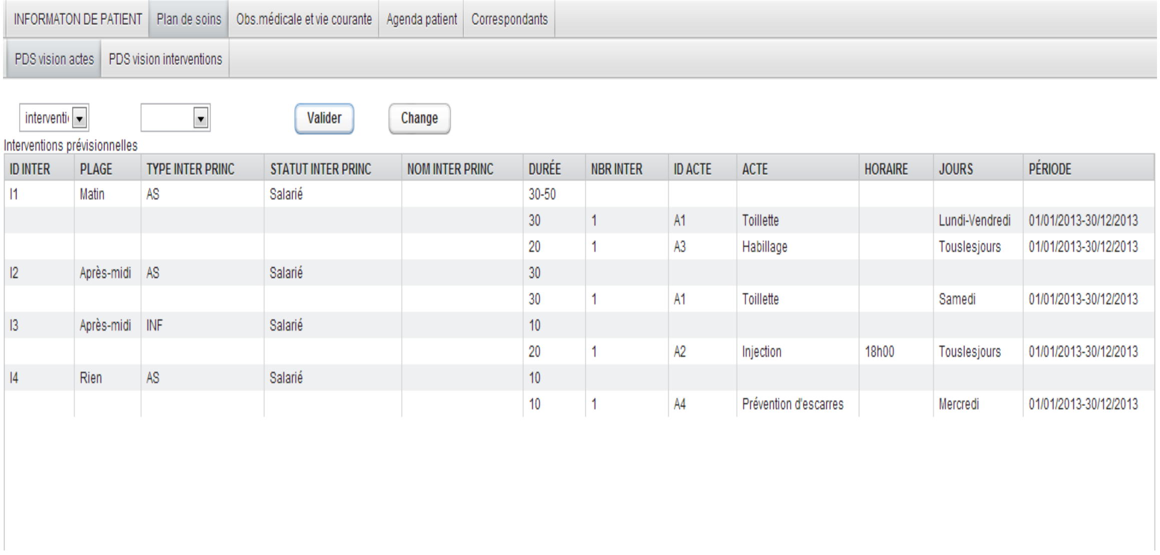


Actes prévus

ID ACTE	CONTROLE	ACTE	TEMPORALITÉS(JOURS--INTERVALLES--PÉRIODE)			TYPE INTER PRINC	NOM INTER PRINC	NBR INTER	DURÉE	REMARQUES
A1		Toilette	Matin Après-midi	Touselesjours Samedi	01/01/2013-31/12/2013 01/01/2013-31/12/2013	AS AS		1	30	
A2		Injection	18h00	Mercredi	01/01/2013-31/12/2013	INF		1	10	
A3		Habillage	Matin	Touselesjours	01/01/2013-31/12/2013	AS		1	20	
A4		Prévention d'e	Rien	Touselesjours	01/01/2013-31/12/2013	AS		1	10	

图 4-7 活动表

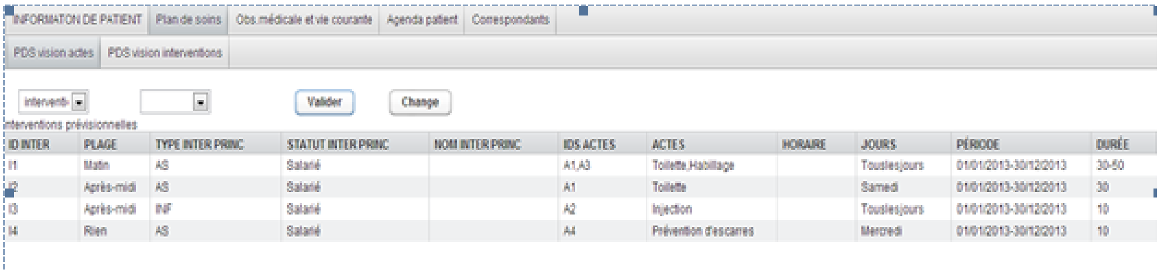
用户点击后按钮“产生参与表”，家庭护理计划将由活动表转化成参与表。用户可以选择两种类型的参与表：完整版参与表和简化版参与表。如图 4-8 所示，参与对象 I_1，它的约束是“时间范围为早上并且护理人员类型为 AS”。参与对象 I_1 包括持续时间为 30 天的 A_1 和持续时间为 20 天的 A_3。所以经计算可得参与对象 I_1 的持续时间为 30-50 天。完整版参与表和简化版参与表见图 4-8，图 4-9。



Interventions prévisionnelles

ID INTER	PLAGE	TYPE INTER PRINC	STATUT INTER PRINC	NOM INTER PRINC	DURÉE	NBR INTER	ID ACTE	ACTE	HORAIRE	JOURS	PÉRIODE
I1	Matin	AS	Salarié		30-50						
					30	1	A1	Toillette		Lundi-Vendredi	01/01/2013-30/12/2013
					20	1	A3	Habillage		Touslesjours	01/01/2013-30/12/2013
I2	Après-midi	AS	Salarié		30						
					30	1	A1	Toillette		Samedi	01/01/2013-30/12/2013
I3	Après-midi	INF	Salarié		10						
					20	1	A2	Injection	18h00	Touslesjours	01/01/2013-30/12/2013
I4	Rien	AS	Salarié		10						
					10	1	A4	Prévention d'escarres		Mercredi	01/01/2013-30/12/2013

图 4-8 完整版参与表



interventions prévisionnelles

ID INTER	PLAGE	TYPE INTER PRINC	STATUT INTER PRINC	NOM INTER PRINC	IDS ACTES	ACTES	HORAIRE	JOURS	PÉRIODE	DURÉE
I1	Matin	AS	Salarié		A1,A3	Toilette,Habillage		Touslesjours	01/01/2013-30/12/2013	30-50
I2	Après-midi	AS	Salarié		A1	Toilette		Samedi	01/01/2013-30/12/2013	30
I3	Après-midi	INF	Salarié		A2	Injection		Touslesjours	01/01/2013-30/12/2013	10
I4	Rien	AS	Salarié		A4	Prévention d'escarres		Mercredi	01/01/2013-30/12/2013	10

图 4-9 简化版参与表

4.4 逻辑层模型检查实现

针对一个家庭护理计划中的活动，我们建立的其时间自动机（见图 4-10）。根据 $A_{activity}=(S,s_0,\sum,X,Inv,T,F)$，我们定义：$S=\{s_0^1,s_1^1,s_2^1\}$ 一个状态集合其中 s_0^1 是穿衣 A_{dress}的初始状态。$F=\{s_2^1\}$ 是最终状态的集合。$X=\{x_d,x_p\}$ 变量的集合，其中 x_d 是测量一天之内的时间而 x_p 衡量整个时期。x_d 和 x_p 都是以小时为单位的。T 表示的是如下的状态转移：

1) $(s_0^1,Activity,x_d\geq 8,\phi,s_1^1)$ 这个状态转移指的是当时间自动机位于 s_0 状态时，可以转移到执行活动的状态 s_1。结合 s_0^1 状态中 $x_d\leq 12$ 和 $x_d\geq 8$ 确保该活动只发生在上午(比如 x_d 在 8 和 12 之间)。

2) $(s_1^1,\varepsilon,x_d==24\wedge x_p\leq 480,\{x_d\},s_0^1)$ 这个状态转移是在特定的时期里($x_p\leq 480$)的每天的结束（x_d=24），时间自动机能从 s_1^1 跳转回 s_0^1 而不需要执行任何活动。在这个状态转移中 x_d 将被设成 0 来记录新的一天。

3) $(s_1^1,\varepsilon,x_p==480,\phi,s_2^1)$ 这个状态转移将被取消在每个时期的介绍（x_p=480）并使自动机转移到 s_2^1 状态种植执行。

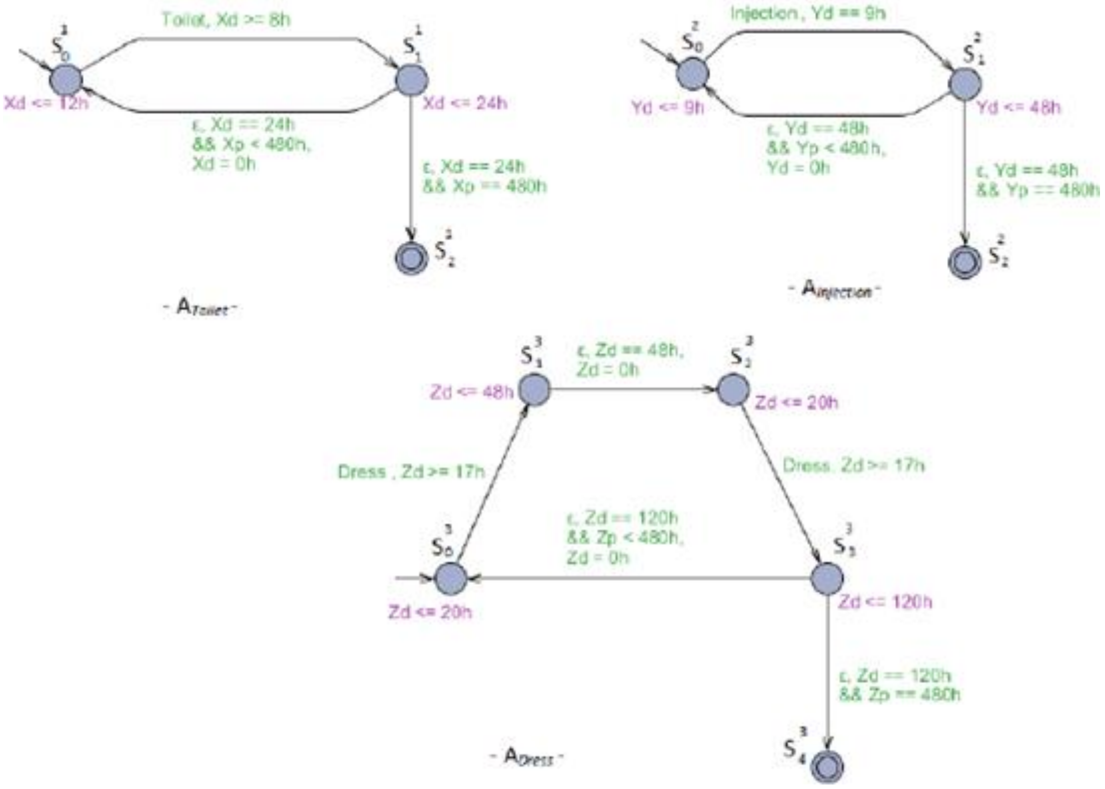


图 4-10 家庭护理计划转化成时间自动机模型

4.5 系统测试

针对用户图形界面的负载性和功能点进行测试。针对用户图形界面，我使用 LoadRunner v9.0 作为测试工具进行测试而针对功能点我使用黑盒测试并设计测试用例。

针对功能点我们设计了功能测试用例。部分测试用例如下：

表 0-1 语言选择测试用例

测试 ID	UI_Test_01
测试描述	用户选择语言并测试是否成功
实测需求	无
测试步骤	(1) 用户选择语言并点击确定键 (2) 用户不选择任何语言并点击确定键
预期结果	如果选择法语，所有信息都以法语显示； 如果选择法语，所有信息都以英语显示； 如果不选择，系统提示“请选择语言”

表 0-2 病人情况选择测试用例

测试 ID	UI_Test_02
测试描述	用户选择病人并验证三种情况
实测需求	成功连接数据库
测试步骤	(1) 用户选择病人信息 (2) 用户创建病人信息
预期结果	如果病人存在，病人的信息将展示给用户； 如果病人不存在，系统提示“没有病人，请创建它”； 如果病人存在但是他的护理计划不完整，系统允许用户再次创建新的家庭护理计划。

表 0-3 活动添加测试用例

测试 ID	UI_Test_03
测试描述	为家庭护理计划添加一个活动
实测需求	成功连接数据库
测试步骤	(1) 选择活动名 (2) 选择时期 (3) 选择重复频率 (4) 选择时间范围 (5) 显示时间区间表并且用户可以拖拽其中的时间区间进行排序 (6) 存入数据库
预期结果	(1) 默认的时期将被数据库提供 (2) 用户可以修改默认的时期 (3) 4 种类型的重复频率可以被选择，其中具体日期一项不能超过时期的范围 (4) 2 种类型的时间范围可以被选择，其中具体时间如果和其他时间范围有冲突，系统将给出提示 (5) 时间区间表有退拽功能进行排序 (6) 时间区间表中的信息顺序将和活动表总时间区间列完全相同

由测试得知，本测试所选取的所有测试用例都成功通过测试并没发现异常。程序运行正常并得到预期结果。

针对系统的负载进行测试，针对本项目需求，我们使用权限法进行系统性能测试。我们针对多人同时进行制定家庭护理计划这一操作进行负载测试，设计如下测试用例：

表 0-4 性能测试用例

测试 ID	PT_Test_05
测试描述	制定家庭护理计划
实测需求	成功连接数据库
测试步骤	(1) 用户选择病人信息 (2) 用户创建病人信息
参与者人数	30
预期结果	制定家庭护理计划事务在最大在线用户数下客户端请求的平均响应时间小于等于 XXs

根据该测试方案对该系统进行测试，以单测试用例中的 30 个并发用户 hiding 家庭护理计划的性能测试为例，其部分测试结果如表 4-5。

表 0-5 性能测试制定家庭护理计划用例的部分测试结果

脚本	Home care.user
总吞吐量/Byte	3115200
总点击量/次	1390
执行时间	00:05:20
平均吞吐量/bps	6648
平均点击率/次/s	2.021

结果分析显示，随着虚拟用户数目增加，制定家庭护理计划事务的响应时间并未发生显著变化，表明该系统能够承受 30 个并发用户的制定家庭护理计划操作。在该场景下，资源图的消耗显示系统的性能也较稳定，满足用户指标。

4.6 小结

本章描述了项目的开发环境并对项目中数据库层，用户图形界面层和逻辑层的实现进行了阐述。之后针对系统的用户界面进行了测试。

结论

本文旨在设计和实现一个新的基于领域特定语言的方法来管理家庭护理计划。首先回顾了现有的管理家庭护理方案的技术如业务流程建模。然而对于护理方案来说，它是一种非结构化的业务流程，所以无法被标准的业务流程建模方法来实现，并且需要医疗领域和数学领域学科之间的配合。为了提高管理家庭护理计划的效率和可行性，本文给出了建议和解决方案。本文提出的这种新型的方法解决了每位患者都需要一个特定的护理计划的问题并且通过创新的家庭护理计划分配工具定义了规范的业务流程。通过分析用户的需求并结合领域特定语言描述家庭护理计划的方法，本文给出了总体的设计。考虑到系统的可扩展性，总体的系统架构分为数据库层，用户图形界面层和逻辑层。数据库层旨在建立一个健壮性的数据库为用户界面层和逻辑层提供准确的数据。用户图形界面层提供一个智能易用的工具来制定家庭护理计划和收集病人和其护理计划的相关数据。逻辑层通过模型检查工具 UPPAAL 生成状态转移图来验证家庭护理计划的合理性。针对就模型分析来说，我们初步地描述和讨论这种使用时间自动机来提供基本的服务来支持分析，验证，制定和管理家庭护理计划的基于 DSL 抽象的方法的初步成果。该系统原型通过测试展现了可喜的成果。结果验证了一种新型的基于领域特定语言在管理家庭护理计划方面的准确性和可靠性。

作为一个刚刚提出的创新型的方法，它需要进行进一步的改进。未来的工作将从以下几个方面展开。首先，我们未来的研究方向将致力于时间自动机的理论框架。我们的初步成果铺平了使用模型检查技术的道路。其次，由于本办法的用户界面仅仅是最终版本的原型，所以界面美化工作功能可以进一步加强，类似触屏和快捷键的定义等功能可以被引进来提高用户体验。

参考文献

[1] R. HÉBERT, A. TOURIGNY et M.GAGNON. Intégrer les services pour le maintien de l’autonomie des personnes – PRISMA (Prog. de Recherche sur l’Intégration des Services pour le Maintien de l’Autonomie) - EDISEM – Canada – 2004.

[2] F. DUCHARM, P. LEBELE et H. BERGMAN. Vieillissement et soins, l’urgence d’offrir des services de santé intégrés aux familles du XX1ème siècle – Ruptures, revue transdisciplinaire en santé, vol. 8, n°2, p. 110-121 – 2001.

[3] NHS Gov.uk, “a care plan” (2012-2014), [Online] Available: http://www.homenursingagency.com (06 August 2012).

[4] P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede and N. Russell. On the Suitability of BPMN for Business Process Modeling. Proceedings of the Fourth International Conference on Business Process Management (BPM 2006), volume 4102, pages 161-176, Vienna, Austria, September 2006. Springer.

[5] Mario Bochicchio, Anna Bruno*, Antonella Longo. Supporting Continuous Improvement in Care Management with BPM. Innovation Engineering Department, University of Salento, Software Engineering. 2011; pp: 32-38.

[6] M. Fowler and R. Parsons. Domain Specific Languages. Addison-Wesley, 2010, pp:15-18

[7] Browne, Eric Donald. Workflow Modeling of Coordinated Inter-Health-Provider Care Plans. Diss. University of South Australia, 2005, page: 12-14.

[8] J. Warren, G. Beliakov, J. Noone, and H. Frankel. Chronic disease coordinated care planning: flexible, task-centered decision support. In Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS-32), vol. 4. Maui, Hawaii, January 1999.

[9] Voutilainen, Päivi, Arja Isola, and Seija Muurinen. "Nursing documentation in nursing homes ﹣ state ﹣ of ﹣ the ﹣ art and implications for quality improvement.”Scandinavian Journal of Caring Sciences 18.1 (2004): 72-81.

[10]Keller, W: Mapping Objects to tables, A pattern language, Proceedings EuroPlop (1997).

[11]Athanasiadis, I.N., Villa, F., Rizzalo, A.E.: Enabling knowledge-based software engineering through semantic-object-relational mappings. 3rd International Workshop on Semantic Web Enabled Software Engineering (2007).

[12]Grönroos, Marko. Book of Vaadin. Vaadin Limited, 2011, pp : 2-5

[13]Bauer, Christian, and Gavin King. Hibernate in action, 2005, pp : 21

[14]Rajeev Alur and David L. Dill. Automata for modeling real-time systems. In Proc. of Int. Colloquium on Algorithms, Languages, and Programming, volume 443 of LNCS, pages 322–335, 1990.

[15]John E. Hopcroft and Jeffrey D. Ullman. Introduction of Automata Theory, Languages, and Computation. Addison Wesley, 2001.

[16]Bengtsson, J., Larsen, K., Larsson, F., Pettersson, P., & Yi, W. (1996). UPPAAL—a tool suite for automatic verification of real-time systems (pp. 232-243). Springer Berlin Heidelberg.

[17]González, Oscar, Rubby Casallas, and Dirk Deridder. "Mmc-bpm: A domain-specific language for business processes analysis." Business Information Systems. Springer Berlin Heidelberg, 2009.

[18]Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: A vision towards using semantic web services for business process management. In: Lau, F.C.M., Lei, H., Meng, X., Wang, M. (eds.) ICEBE, pp. 535–540. IEEE Computer Society, Los Alamitos (2005)

[19]Miers, D., Harmon, P., Hall, C.: The 2007 bpm suites report – a detailed analysis of bpm suites version 2.1, Business Process Trends (2007), http://www.bptrends.com/

[20]Lau, C., Peddle, S., Yang, S.: Gathering monitoring metrics to analyze your business process, IBM (December 2007), http://www.ibm.com/us/

[21]P. Dadam, M. Reichertand, and K. Kuhn. Clinical workows - the killer application for process-oriented information systems Business, pages 1{15, 2000.

[22]R. Alur. Timed automata. In N. Halbwachs and D. A. Peled, editors, Proceedings of the 11th International Conference on Computer Aided Verification (CAV'99), volume 1633 of Lecture Notes in Computer Science.

[23]J. Bau, E. Bursztein, D. Gupta, and J. C. Mitchell, "State of the Art: Automated Black-Box Web Application Vulnerability Testing," in IEEE Symposium on Security and Privacy, 2010, pp. 332-345.

[24]Jesse James Garrett. (2005) Adaptive Path. [Online]. http://www.adaptivepath.com/publications/essays/archives/000385.php

[25]K. Benjamin, G.v Bochmann, G.V Jourdan, and I.V Onut, "Some Modeling Challenges when Testing Rich Internet Applications for Security," in First International workshop on modeling and detection of vulnerabilities (MDV 2010), Paris, 2010.

[26]A. Arasu, J. Cho, A. Garcia-Molina, A Paepcke, and S Raghavan, "Searching the web," ACM Transactions on Internet Technology, vol. 1(1), pp. 2-43, 2001.

[27]S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," Computer Networks and ISDN Systems, vol. 30(1-7), pp. 107-117, 1998.

[28]Cirilo C E, do Prado A F, de Souza W L, et al. Building Adaptive Rich Interfaces for Interactive Ubiquitous Applications [J].

[29]Fowler M. Domain-specific languages [M]. Pearson Education, 2010.

[30]D. Amalfitano, R. Fasolino, and P. Tramontana, "Rich Internet Application Testing Using Execution Trace Data," in Proceddings of Third International Conference on Software Testing, Verification, and Validation Workshops , Washington, DC, USA, 2010, pp. 274-283.

[31]M. Bouet, K. Gani, M. Schneider, and F. Toumani. A general model for specifying near periodic recurrent activities - application to home care activities. Technical report-LIMOS, 2013.

[32]Kamara Benjamin, Gregor von Bochmann, Mustafa Emre Dincturk, Guy-Vincent Jourdan, and Iosif Viorel Onut, "A Strategy for Efficient Crawling of Rich Internet Applications," in Web Engineering: 11th International Conference, ICWE 2011,Paphos, Cyprus, Sören Auer, Oscar Dáz, and George Papadopoulos, Eds.: Springer Berlin / Heidelberg, 2011, vol. 6757, pp. 74-89.

[33]Antonsson M. Modeling of real-time systems in UML with Rational Rose and Rose Real-Time based on RUP [J]. 2001.

[34]Fern ́andez-Luque, F., Zapata, J., Ruiz, R., Iborra, E.: A wireless sensor network for assisted living at home of elderly people. Lecture Notes in Computer Science 5602 (2009) 65–74

[35]Dong J S, Hao P, Qin S, et al. Timed automata patterns [J]. Software Engineering, IEEE Transactions on, 2008, 34(6): 844-859.

[36]Bengtsson, J., Yi, W.: Timed automata: Semantics, algorithms and tools. In:

Lectures on Concurrency and Petri Nets. (2003) 87–124

[37]Tripakis, S., Dang, T.: Modeling, verification and testing using timed and hybrid automata. In Mosterman, P., Nicolescu, G., eds.: Model-Based Design for Embedded Systems, CRC Press (2009) 383–436

[38]Bot´ıa, J.A., Villa, A., Palma, J.T.: Ambient assisted living system for in-home monitoring of healthy independent elders. Expert Syst. Appl. 39(9) (2012) 8136–8148

[39]Lientz B P, Swanson E B. Software maintenance management: a study of the maintenance of computer application software in 487 data processing organizations [J]. 1980.

[40]Morris, John N., et al. "Comprehensive clinical assessment in community setting: applicability of the MDS-HC." Journal of the American Geriatrics Society 45.8 (1997): 1017-1024.

[41]Kempe, Allison, et al. "Quality of care and use of the medical home in a state-funded capitated primary care plan for low-income children." Pediatrics 105.5 (2000): 1020-1028.

[42]Hui-li, Zhang, et al. "Research of Load Testing and Result Application Based on LoadRunner." (2012).

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于DSL的家庭护理计划管理方法及工具设计与实现 DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：吴元吟　　　　日期：　2013 年 8 月 14 日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1)学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：吴元吟　　　　日期：　2013 年 8 月 14 日

导师签名：（代）　　　　日期：2013 年 11 月 26 日

致谢

我想表达对我的硕士论文导师 Farouk Toumani 教授深深的谢意。感谢他提供给我实习工作并在实习阶段给我耐心地指导。成为 Farouk Toumani 教授的学生，我学到很多课本外的知识。我同样感谢来自哈尔滨工业大学的聂兰顺老师和来自 ISIMA 的 Kun-Mean Hou 教授。感谢他们花费大量时间阅读我的论文并给予我建设性的意见。他们的博学，热情地助人为乐精神和严谨的治学态度给我留下了深刻的影响。

特别感谢哈尔滨工业大学软件学院和 ISIMA 对我的培养。此外，我要感谢所有老师对我们全心全意地教授知识。

同时，我也要感谢我们小组的成员，特别是 Sneijder Michel 教授，M. Bouet 教授和 Gani Kahina。你们是我的良师益友。没有你们的帮助，我根本无法完成此次的项目。

个人简历

吴元岭

19, Rue Roche Gen ès - 63173 Aubi ère C édex, France
+33(0)641674992
wuyuanling555@163.com

Profile

本人致力于业务流程建模软件开发和数据库研究，有两年以上的网页开发和应用程序设计实习经历。有良好的团队合作精神和文档书写能力。

教育背景

2011.9-至今　哈尔滨工业大学，克莱蒙费朗二大-计算机科学，软件工程-硕士研究生学位,双学位
2007.9-2011.7　哈尔滨工业大学-软件工程-学士学位

项目经验

- **基于领域特定语言原理来管理家庭护理计划的方法的设计与实现（实习）**
  Le LIMOS (Laboratoire d’Informatique, de Modélisation et d’Optimisation des Syst èmes)
  2013 年 4 月 – 现在 | 法国克莱蒙费朗
  描述：基于领域特定语言原理来管理家庭护理计划的方法的设计与实现. 这种新方法用来替代传统的管理方案并提高制定家庭护理计划的可行性。
- **代码复查管理系统，负责软件质量保证和学生课程管理（实习）**
  软件学院实验室 (哈尔滨工业大学)
  2010 年 10 月 – 2011 年 7 月 (10 个月)
  描述：设计和实现一个代码复查平台来保证代码的规范和质量。
- **sina micro-blogging application, responsible for data collection, data mining and social network (Internship)新浪微博应用，社交网络商业价值的深度挖掘（实习）**
  软件学院实验室 (哈尔滨工业大学)
  2011 年 10 月 –2012 年 1 月 (4 个月)
  描述：基于新浪微博平台的应用开发，通过构建社交网络人际关系图，来深度挖掘其商业价值

本人特质

我有较强的沟通能力和管理能力，和良好的人际关系。作为一个快速的学习者和勤奋的人，我喜欢来自技术和生活中的各种挑战和困难并有足够的勇气来解决这些问题。我相信，这种变化有利于有准备的头脑。

个人技能

- 拥有2年以上的项目领导经验以及4年以上的商业软件设计与开发经验。
- 精通**ERP**以及企业集成和建模领域的相关知识与技术。
- 拥有4年以上应用**.NET**（**C#**）以及 **Java**进行面向对象、多线程软件开发的经验。
- 拥有使用**ASP.NET**、**Eclipse** （**J2EE**）以及**JavaScript**（**AJAX**）相关技术进行网络应用软件的丰富开发经验。
- 语言能力：英语（托业 770）、中文、法语。

所学课程

Advanced Mathematical Analysis, Algebra and Geometry, Set Theory & Graph Theory, Probability Theory & Stochastic Process, Mathematics Modeling; C Language, Java, C++, .Net Framework; Computer Organization, Operating System, Principles of Compiler, Embedded System, Computer Networking, Graphical User Interface, Distributed Systems; Data Structure & Algorithms, Database, Software Component, Software Architecture, J2EE, Web Services; Software Engineering, Software Quality Assurance and Testing, System Analysis and Design.

硕士学位论文

Dissertation for Master’s Degree

(工程硕士)

(Master of Engineering)

基于 DSL 的家庭护理计划管理方法及工具的设计与实现

DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS

吴元岭

哈爾濱工業大學

Université Blaise Pascal

2013 年 9 月

国内图书分类号：TP311 学校代码：10213
国际图书分类号：681 密级：公开

工程硕士学位论文
Dissertation for the Master’s Degree in Engineering
(工程硕士)
(Master of Engineering)

基于 DSL 的家庭护理计划管理方法及工具的设计与实现

DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS

硕士研究生：	吴元岭
导师：	聂兰顺、副教授
副导师：	Kun Mean HOU、教授
实习单位导师：	Farouk Toumani、教授
申请学位：	工程硕士
学科：	软件工程
所在单位：	软件学院
答辩日期：	2013 年 9 月
授予学位单位：	哈尔滨工业大学

Classified Index: TP311
U.D.C: 681

Dissertation for the Master’s Degree in Engineering

DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS

Candidate:	Yuanling Wu
Supervisor:	Lanshun Nie Associate Profeesor
Associate Supervisor:	Kun Mean HOU Proffessor
Industrial Supervisor:	Farouk Toumani Professor
Academic Degree Applied for:	Master of Engineering
Speciality:	Software Engineering
Affiliation:	School of Software
Date of Defence:	September, 2013
Degree-Conferring-Institution:	Harbin Institute of Technology

摘 要

家庭护理是指由医护人员在患者的家中提供卫生保健或支持性护理。近些年来，家庭护理正在广泛地开展起来，尤其是在人口老龄化严重的西方社会。同时，管理家庭护理计划面临严峻的挑战。

在一般情况下，传统管理家庭护理计划的方法无非是通过纸质文档记录在案或者使用业务流程建模（BPM）技术进行管理。然而，家庭护理计划可以看作是一种非结构化业务流程，因此它不适合使用在医疗领域中标准的业务流程建模技术，并且流程建模管理病不是一件容易的事。因此，我们提出了一种新型的基于 DSL（领域特定语言）的方法，通过使用高层次和面向用户的抽象来描述和管理家庭护理计划。考虑指定家庭护理计划的合理性，我们提出的领域特定语言是一种图形语言，它提供了适合家庭护理受益者的基本结构。首先通过从元模型到数据库的转换，我们获取稳定的家庭护理计划存储载体。基于此，我们使用 Vaadin 的框架来实现一种简化制定护理计划智能工具。最后，我们集合基于时间的自动机形式化描述领域特定语言来分析和描述初步结果以支持分析，验证和管理家庭护理计划。

该项目原型通过终端用户测试展示了可喜的成果。结果验证了基于DSL（领域特定语言）的方法是更合适和准确的来管理家庭护理计划。

关键词：领域特定语言，家庭护理计划；定时自动机， Vaadin 的框架

Abstract

In recent years, home care has been a real craze. It refers to health care or supportive care delivered by health care professionals in patients' homes, especially in the Western societies which are distributed to the aging population. Meanwhile, managing home care plans meets serious challenges. Our project aims to solve the problem how to manage home care plan.

In general, the traditional methods to manage home care plan are documents in the paper record or BPM (Business Process Modeling) technique. However, home care plans can be viewed as a kind of non-structured business processes which cannot be managed using standard BPM technology and process modeling in the medical field is in general not an easy task. Hence, we put up with a novel DSL (Domain Specific Language) based approach tailored to express home care plans using high level and user-oriented abstractions. Considering of reasonableness for specifying home care plan, the proposed DSL is a graphical language which provides basic constructs suitable for home care stakeholders. Through transformation from meta-model to relational database, we use vaadin framework to implement an intelligent tool to simplify the process of making care plan. Finally, we describe and discuss preliminary results regarding formalization of the proposed DSL abstractions using timed automata in order to provide basic services to support analysis, verification, enactment and management of home care plans.

The project prototype demonstrated promising results of the approach for the end user testing. The results validated the DSL (Domain Specific Language) based approach is more suitable and accurate to manage home care plans.

Keywords: Domain Specific Language; Home Care Plan; Timed Automata; Vaadin Framework

Résumé

Les soins à domicile correspondent à l’ensemble des soins ou des services pratiqués par des personnels soignants ou des prestataires privés d'assistance médicale au domicile d’un patient. Au cours des dernières années, les soins à domicile ont connu un véritable engouement particulièrement dans les sociétés occidentales où la population est de plus en plus vieillissante.

Actuellement, la gestion des plans de soins à domicile pose de nombreux problèmes. Traditionnellement, des méthodes s’appuyant sur des documents papiers ou sur la technique BPM (Business Process Modeling) sont utilisées. Or, les plans de soins à domicile peuvent être considérés comme une sorte de processus metier non-structurés qui ne peuvent pas être gérés en utilisant la technologie BPM standard. C’est pourquoi, nous avons mis en place une nouvelle approche basée sur la notion de DSL (Domain Specific Language) qui permet d’exprimer les plans de soin à l'aide d’un language de haut niveau d’abstraction.

La notion de DSL est un langage graphique qui fournit des constructions de base appropriées pour les acteurs de soins à domicile. Grâce à la transformation du méta-modèle de base de données relationnelles, nous utilisons le framwork Vaadin pour implémenter un outil intelligent afin de simplifier le processus de construction du plan de soins. Enfin, nous décrivons et discutons des résultats préliminaires concernant la formalisation des abstractions DSL proposées en utilisant des automates temporisés afin de fournir des services de base qui vont faciliter l'analyse, la vérification, l'adoption et la gestion des plans de soins à domicile.

Les premiers résultats expérimentaux obtenus sont très prometteurs. Ils confirment le fait que l’approche DSL (Domain Specific Language) est particulièrement efficace pour gérer au mieux les plans de soins à domicile.

Mot-clés: Langage de domaine spécifique; Automates temporisés; Régime de soins à domicile; Framework Vaadin

目　录

Chapter 1 Introduction

1.1 Background

In recent years, home care (also referred to as domiciliary care) has been a real craze. This is explained by the fact that a major concern of Western societies is the management of the aging population. Indeed, the elderly frail people are vulnerable and often have chronic, who require continuous and long-term care. Furthermore, letting these people in their ordinary circumstances and contacting with their social networks is a key to avoid depression or exclusion, especially including other factors such as children, families, hospital saturation, etc...[1][2] Thus, in this case, new structures such as Home Hospitalization, Home Nursing and Home Maintaining emerge as the times require. While Home Hospitalization provides supportive care or health care to patients with serious diseases, Home Nursing offers patient education, innovative technologies, and specialized pediatric and adult services. As for Home Maintaining, it brings to the drug addicts that nurses help them stay at home as long as possible.

The trend today is to consider the care of a patient in a unified way, even if it passes from one structure to another. To ensure a good organization, quality and effectiveness of home care, organization of care coordinators and different treatments will be reflected in the form of care plan. Home care refers to health care or supportive care delivered by health care professionals in patients' homes. The target objective for social and economic reasons is to enable people that require care to remain at home instead of having, long-term, stays in hospitals or health care facilities. Several types of care may be provided to persons in their own homes including health services, e.g., hospital-level care, and activity of daily living such as bathing, dressing, using the toilet, etc. These services are delivered by an interdisciplinary care team. A care plan defines all the services provided for a given person and coordinates the involved health care professionals. Such a plan is usually constructed through a complex process involving a comprehensive assessment of patient's needs (e.g., medical, nursing, social needs) as well as its social and physical environment. A care plan is a key element in the process of home care. A care plan is an agreement between you and your health professional (and/or social services) to help you manage your health day-to-day. It can be a written document or something recorded in your patient notes [3].

In this context, our project is a part of Plas'O'Soins. Plas'O'Soins is a collaborative research project funded by the ANR (Agence Nationale de la Recherche française) and coordinated by ISIS. The project objective is to provide a platform to help monitoring and coordination of home care. This project is funded as part of the call for projects ANR TecSan 2010, for a term of three years beginning in April 2011 and has been approved by the Competitiveness Cluster Cancer-Bio-Santé. The consortium Plas'O'Soins project combines academics, industrialists and Health ICT field and end users. Associated partners contribute to project management, ethical and legal aspects, and also represent guardianship that finance PAD in France. Plas'O'Soins deals with the coordination of interventions, continuity of care and traceability activities within the supported at home. The project will provide a platform to identify and control processes agile and customized care, assisting in the coordination and planning of interventions, while facilitating communication between stakeholders and providing dashboards to quantify the efficiency of the device.

My project is interested in the problems faced by coordinators to establish care plans as a support. The ultimate goal is to offer an easy-to-use and feasible approach to manage home care plans.

Through a thorough review of the various anonymous documents, we used UML (Unified Modeling Language) to establish our model in the form of class diagrams. This model has allowed us to identify key concepts revolving around a care plan as acts, general acts, time, connectors (parallel activities in sequence), etc... However, other than general model, coordinators who are responsible for managing and monitoring home care plans among different actors need specify the various home care plans to different patients. So we decide to use a DSL (domain-specific language)-based approach to solve this problem instead of traditional business process management method. For our domain-specific language of home care plans, we define that it is the set of objects that may be involved in a plan of care and handling that can be done on these objects so that we describe the care plan management process by our approach.

1.2 The purpose of project

The title of my project is “Design and implementation of a DSL-based approach to manage home care plans”, which is provided by LIMOS (Informatique, Modélisation et Optimisation des Systèmes). As part of the project Plas'O'Soins, we are interested by the problems underlying the design and management of home care plans. My project aims to offer a novel DSL (domain specific language)-based approach to manage home care plans and developed a tool for visualization which is easy to express, monitor and advance the process of care management rapidly and exactly. The cornerstone of our approach lies in the definition of a DSL (Domain Specific Language), or more specifically a business DSL, tailored to express home care plans using high level abstractions. The proposed DSL is a graphical language which provides basic constructs suitable for home care stakeholders.

1.3 The status of related research

1.3.1 Home Care Plan

In the term of “care plan”, it is quite broad, encompassing virtually term that describe the processes required to treat the clinical condition of a single patient over an extended period of time. It is normally developed to deal with chronic conditions or conditions involving a multidisciplinary team of carers [7].

In home care plan, patients who want to get healthcare send their requirements to hospital or community which will specify the nurse to evaluate the condition of patient and supply a list of service related to care plan. After the patient accept the agreement, doctor coordinators start to make a detailed and personal home care plan in which there are acts, the information of nursing stuff, schedule and so on. The general home care plan process is shown in figure 1-1.

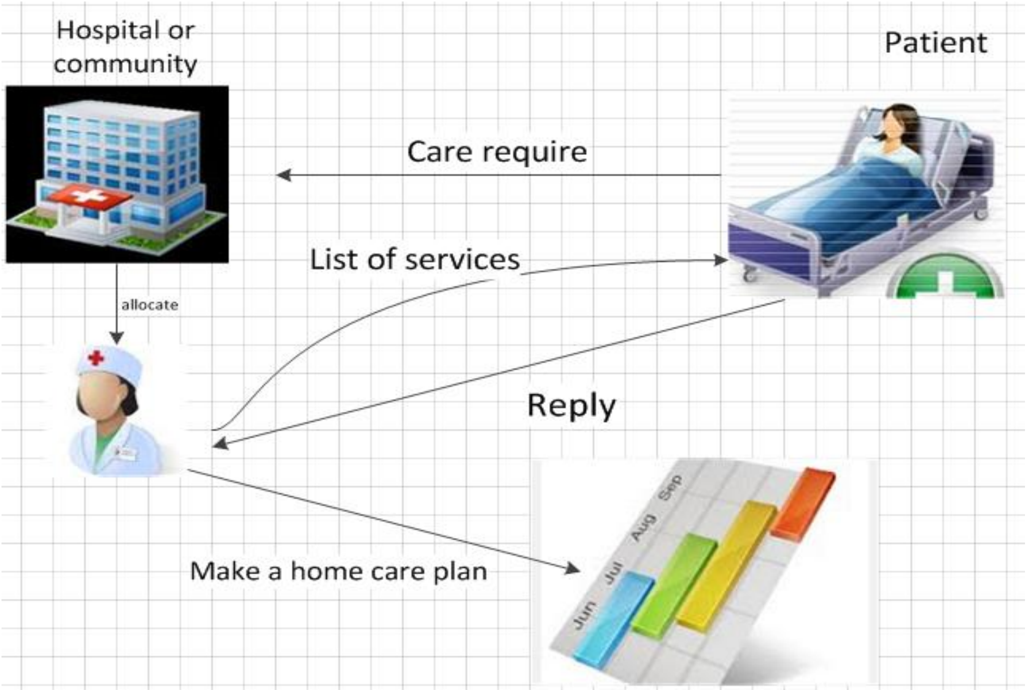
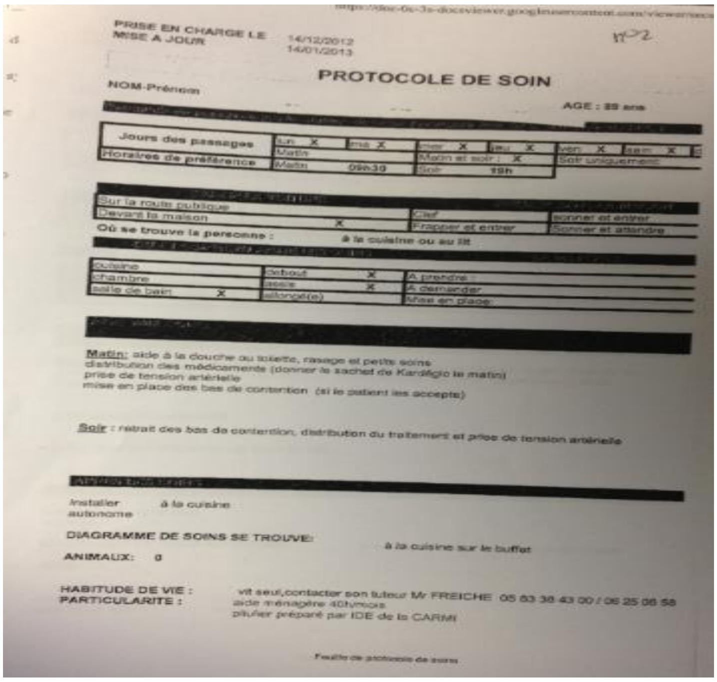


Figure 1-1 care plan process

In normal circumstances, coordinators prefer to manage daily care plan by the written description such as papers or documents which are shown in Figure 1-2, even if this mode is much more complicated in the beginning. With the growing population of patients who accept home care plans, which has the effect on the increasing number of care plans, the written descriptions of the care plans become difficult to manage, such as the difficulty to adjust or update. Another consequence is that the care plan documents often lack formal structure, so that coordinators are not able to edit a clear version of those documents to his satisfaction. What is more, for those working in long term care, accumulated nursing knowledge about their patients are not always drawn on paper. In addition, if the information is inaccurate or inappropriate, it will be obvious to go against patients' safety and the ongoing care plans.



PRISE EN CHARGE LE 14/12/2012
MISE A JOUR 14/01/2013

N°2

PROTOCOLE DE SOIN

NOM-Prénom

AGE : 89 ans

[illegible]

Jours des passages	lun X	ma X	mer X	jeu X	ven X	sam X	d
Horaires de préférence	Matin		Matin et soir : X		Soir uniquement		
	Matin 09h30		Soir 19h				

[illegible]

Sur la route publique		
Devant la maison X	Clef	sonner et entrer
	Frapper et entrer	Sonner et attendre

Où se trouve la personne : à la cuisine ou au lit

[illegible]

cuisine		
chambre	debout X	A prendre :
salle de bain X	assis X	A demander
	allongé(e)	Mise en place

[illegible]

Matin: aide à la douche ou toilette, rasage et petits soins
distribution des médicaments (donner le sachet de Kardégic le matin)
prise de tension artérielle
mise en place des bas de contention (si le patient les accepte)

Soir : retrait des bas de contention, distribution du traitement et prise de tension artérielle

[illegible]

Installer à la cuisine
autonome

DIAGRAMME DE SOINS SE TROUVE: à la cuisine sur le buffet

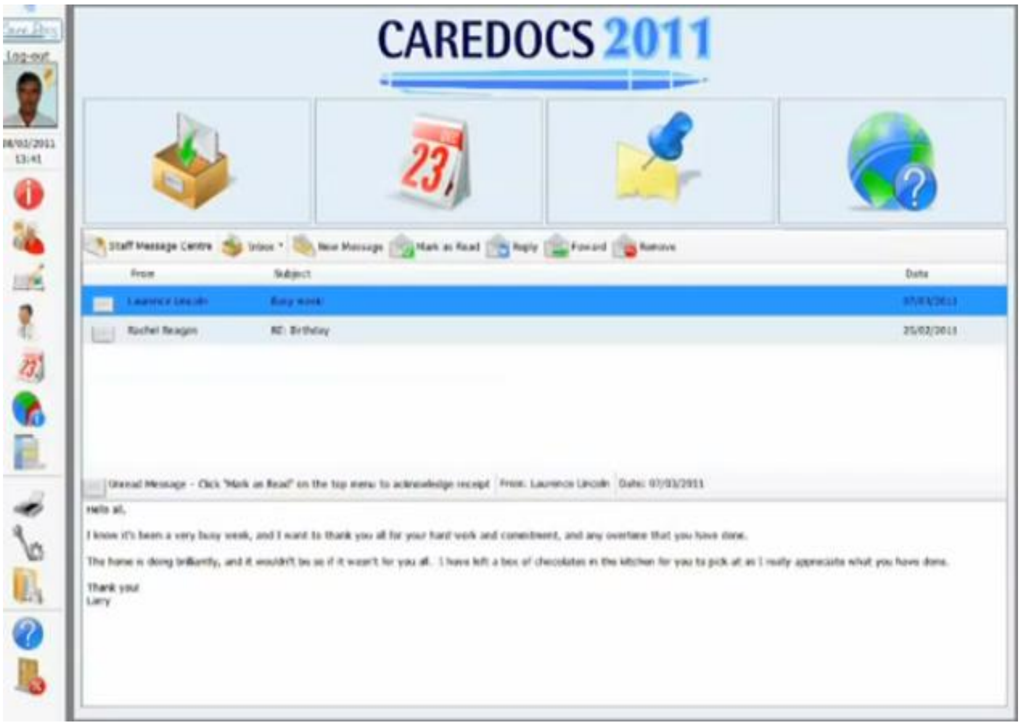
ANIMAUX: 0

HABITUDE DE VIE :
PARTICULARITE :
vit seul,contacter son tuteur Mr FREICHE 05 63 38 43 00 / 06 25 08 58
aide ménagère 40h/mois
pilulier préparé par IDE de la CARMI

Feuille de protocole de soins

Figure 1-2 documents of care plans

To solve the problem as we mentioned above, several computer-based tools or systems appeared recently such as CareDocs. CareDocs is an innovative computer-based care planning and home management system for Residential and Nursing Care Homes. Designed by Care Home owners, CareDocs offers all the features that you'll need to ensure the smooth running of your Care Home. Care planning is at the heart of CareDocs and as well as providing a quick and easy way to carry out comprehensive assessments, you'll be able to produce fully private care plans in a matter of minutes. Managers have immediate access to all the information about residents, staff and the care home, presented on a single page. Reminders are automatically generated to ensure that all records are up to date and all aspects of home management are not overlooked. However, we find that the CareDocs system only permits coordinators to input information in order to replace paper documentation. It is not able to help coordinators arrange care plan and meet all requirements of no-structured home care plan. CareDocs is shown in the Figure 1-3.



From	Subject	Date
Laurence Lincoln	Busy week	07/03/2011
Rachel Reagan	RE: Birthday	25/02/2011

Unread Message - Click "Mark as Read" on the top menu to acknowledge receipt | From: Laurence Lincoln | Date: 07/03/2011

Hello all,

I know it's been a very busy week, and I want to thank you all for your hard work and commitment, and any overtime that you have done.

The home is doing brilliantly, and it wouldn't be so if it wasn't for you all. I have left a box of chocolates in the kitchen for you to pick at as I really appreciate what you have done.

Thank you!
Larry

Figure 1-3 the existing software CareDocs

Good patient management information systems, network infrastructure and efficient communication systems are all prerequisites for efficient management of inter-provider care plans. The Australian Coordinated Care Trials (Round 1: 1997-1999; Round 2: 2002-) were used by a number of participants to test prototype care planning methodologies. In the South Australian HealthPlus Coordinated Care Trial, an online facility for presenting care plans via browser technology was trialed [7]. The software, Care Plan On-Line (CPOL) [8] assisted practitioners to tailor a “base” care plan for an individual, by augmenting their own knowledge and experience with additional computer supplied knowledge from a range of sources. The result was demonstrated that the system that helps doctors assign medical home care plan become efficient with efficiency of network infrastructure improved. However, a set of format standards of home care plan for all the participants, namely patients, would not be put forward.

1.3.2 Business Process Modeling

The other idea is that Business Process Management is applied in manage home care plans, and in particular Business Process Modeling Notation (BPMN), as the tool used for the detailed modeling of the care plan management process, and for further analysis of the results. This idea is also supported by other authors and studies. Recently, in fact, a novel approach was proposed, called Use Process [4], which combines Business Process Modeling Notation (BPMN) and UML Use Case Diagrams in order to model requirements. The research position of these authors takes into account that the successful operation of the enterprises is based on their business processes, so they state that a Business Process-oriented approach to gather requirements can lead in a strong involvement of the customer in the requirements definition, and as a result a successful project. They developed the Use Processes (UP) approach to gather requirements, which consists of a Use Process Diagram (UPD) based in notation elements of the BPMN v2.0 and the Use Case Diagrams (UCD) of UML v1.5. Also they presented a set of templates to describe the elements of a UPD. The purpose of the UPD is to present a general view of the functionalities that a system must provide within the activities of a BP [9].

The approach adopted by the research group to assess the Leonardo project was based on the following phases: first they have modeled the care management process as implemented in Leonardo Project (AS IS model) using BPM [4].AS IS model is shown in figure 1-4. Then they have analyzed the software package InformaCare supporting the care management process by identifying user profiles and functionalities; then they have analyzed the Key Performance Indicators identified by Pfizer for the care management process and finally they have conducted targeted interviews and questionnaires to CMs regarding the process and InformaCare, in order to trace a new Business Process model associated to the improved process (TO BE model), elicit the new tool's requirements and define performance indicators in order to control the whole system[4].

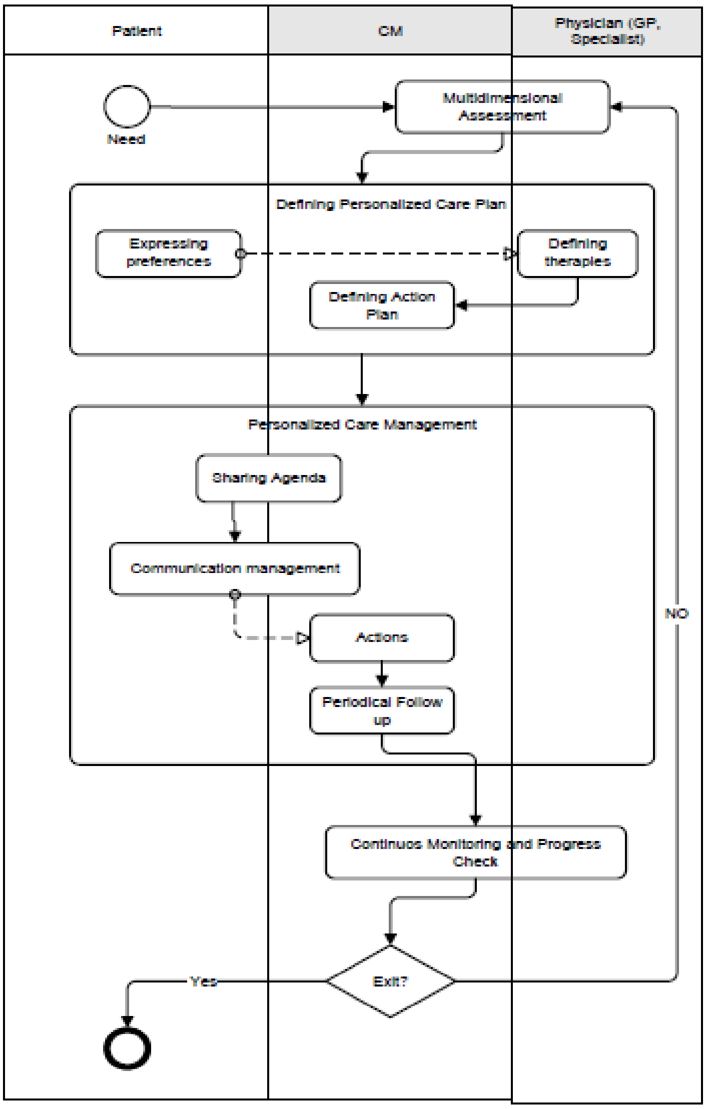

Patient
CM
Physician (GP, Specialist)
Need
Multidimensional Assessment
Defining Personalized Care Plan
Expressing preferences
Defining therapies
Defining Action Plan
Personalized Care Management
Sharing Agenda
Communication management
Actions
Periodical Follow up
NO
Continuos Monitoring and Progress Check
Yes
Exit?

Figure 1-4 BPMN-like model of Care Management Process in Leonardo Project (AS IS) [4]

We quickly found that modeling care plan process by using BPMN did not meet all the goals. Although most of the concepts of a care plan can be modeled in BPMN, it fails to fully and easily represent all forms of time constraints or process-related home care frequencies. In addition, the modeling obtained some complexity and is not easily handled by the coordinators. That is why we want to keep BPMN to represent the process of home care but that we explore in the way of more specific languages. The DSL, Domain Specific Language (DSL), are programming languages or computer modeling limited expressiveness, but specific to a particular area [6]. DSL are already widely used in the fields of computer science and mathematics, have recently coveted in the field of health. For now, we direct our work over to an internal DSL (not external), that is to say, a language based on a main language in our case could be BPMN.

Some researches show that BPM evidences the difficulty of analyzing the execution of processes from a business perspective [18]. Firstly, while the process models are automated from a business perspective in a workflow implementation, the analysis specifications are realized from an implementation perspective. Although there are many tools and techniques to analyze business processes (e.g., business activity monitoring) [19] [20], typically these solutions use their own languages to encode analysis concerns in the implementation of the workflow system (e.g., BPEL or java). Secondly, typically these analysis specifications result in tangled and scattered knowledge in the process code. These low-level mechanisms decrease the maintainability and reusability capabilities and increase the complexity because analysis concerns have to be repeated or adapted every time the process changes. On the other hand, current BPA solutions typically only perform analysis in terms of process execution information (e.g., time running, current state) but not in terms of the inner definition of basic activities (data). However, it is necessary to be aware immediately when a critical attribute value is assigned in order to take complementary decisions (e.g., allow data modification). They define a domain-specific language to specify analysis concerns in terms of domain concepts (i.e., BPMN) and a mechanism to express the data used by the process model to improve analysis capabilities and provide the execution semantics of our language with a suitable aspect-oriented workflow language. Its approach helps in making the measurement variables and the associated rules more transparent to the end user by explicitly modeling them with a domain specific language and within an integrated process data model [17].

So by the BPMN method, managing home care plans becomes challenging for several reasons. First, process modeling in the medical field is in general not an easy task [25]. Indeed, medical processes require usually complex coordination and interdisciplinary cooperation due to involvement of actors from various health care institutions. Moreover, home care plans display the following features that make them difficult to handle with traditional BPM technologies:

1) A home care plan can be viewed as a collection of repetitive activities, (e.g., medical acts) which are repeated during a given period. The activities are however enacted according to an irregular schedule. The irregularities of an activity have two

main causes. At first, the activity very often has to follow the evolution of the needs which is usually irregular. This type of irregularity is characterized by strengthening or weakening in the rhythm of realization. Then, an activity which requires human resources has to respect life cycles appropriate to this type of resources and in particular rest time of the weekend. This type of irregularity induces interruptions in the rhythm of the realization. Specification of irregular activities requires the use of suitable temporal constraints.

2) Home care plans are essentially unstructured processes in the sense that each patient has its own specific care plan. Indeed, the care plan for each patient is developed on an individual basis because each patient is unique whether at its pathology or needs. Therefore, it is simply not possible to design a unique process capturing in advance the care plans of all the patients. In other words, a traditional approach model once, execute many times" is not sustainable in our context.

1.3.3 Vaadin Framework

For our project, most of frameworks are chosen to consistute the intelligent tool for coordinators, such as web sites (PHP, JSP), desktop application and mobile application. Based on principles which are to simplify the user's operation and meet the requirements of specifying home care plans, as well as being convenient for developers, we chooses Vaadin which combines websites with application as our framework. Vaadin is an AJAX web application development framework that enables developers to build high-quality user interfaces with Java, both on the server- and client-side. It provides a set of libraries of ready-to-use user interface components and a clean framework for creating your own components. The focus is on easy-of-use, reusability, extensibility, and meeting the requirements of large enterprise applications. Vaadin Runtime Architecture is described in Figure 1-5. We will use Vaadin framework to design GUI, both client-side and server-side [12].

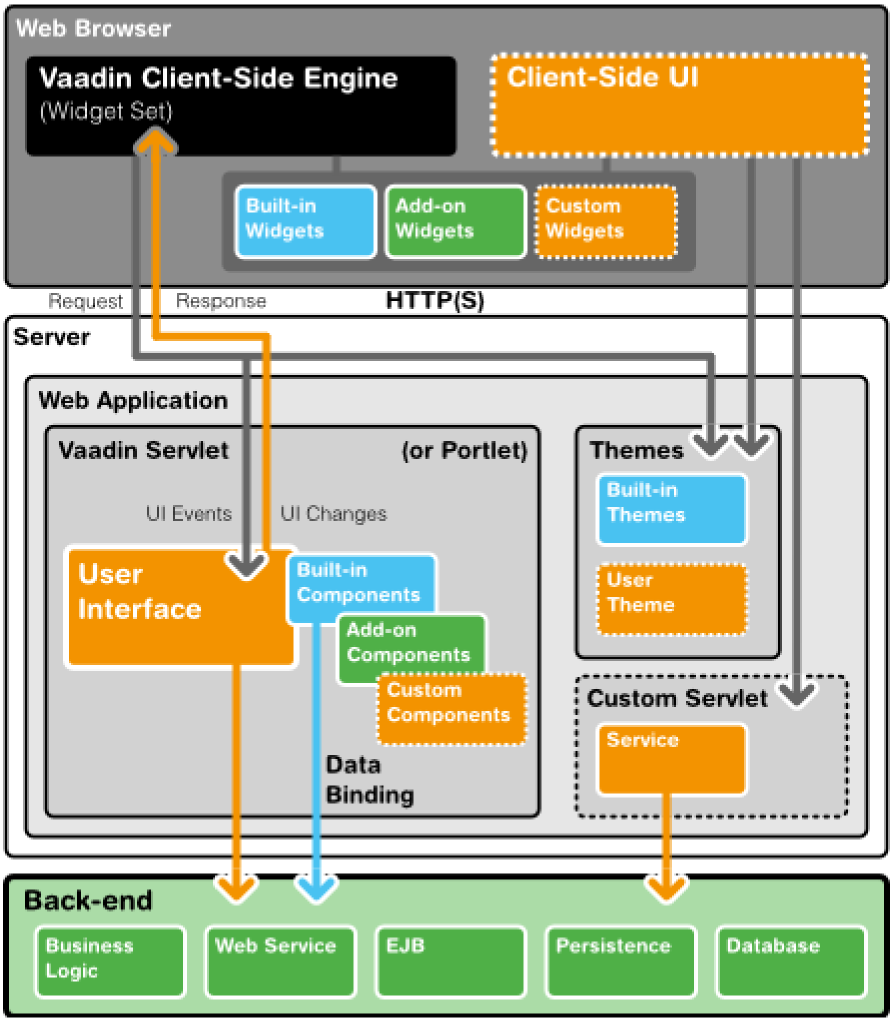


Figure 1-5 Vaadin runtime architecture

"Server-Side Vaadin Application Architecture" illustrates the basic architecture of server-side web applications made with Vaadin. This architecture consists of the server-side framework and a client-side engine that runs in the browser, rendering the user interface and delivering user interaction to the server. The user interface of the application runs as a Java Servlet session in a Java application server, and the client-side engine as JavaScript.

The main advantages of Vaadin is comparing to another more settled RIA frameworks are as follows: You can manage the whole UI with Java code. These are great news because developers don't need to learn every single aspect from another RIA frameworks such Struts or Java Server Faces. It is a truth that there are more

Struts or JSF developers than Vaadin ones, but it is a truth so that there are more people who can write Java code than JSF or Struts components.

This initial disadvantage is solved because the next two reasons: To begin with, The Vaadin developer team has built a great number of core UI components for Vaadin. We can take a look clicking in vaadin demo samplers. The second reason is that developed by Vaadin, developers could need design CSS or HTML5 to landscape the user interface because all the user interface design would be generated automatically. Meanwhile, Vaadin framework supplies a lot of themes for developers to replace JavaScript.

1.3.4 Timed Automata

In order to check the reasonableness of home care plans, we use timed automata to verify our model. Timed automata were introduced in the 90s by Rajeev Alur and David Dill. This formalism is an extension of finite state automata with variables of type clock [21]. Indeed, a timed automaton is defined by a set of nodes (states), transitions, actions and clocks. The nodes represent the states of the system, the transitions are labeled by event symbols, and clocks are variables with zero or positive real values.

While state transitions are instantaneous, in every state, the time may elapse; in fact, the clocks are used to define constraints on the transitions and/or states. Constraints on states are called invariants, they allow to specify that the system can remain in the state while the constraint is satisfied. As for the constraints associated to transitions, called guards, they allow to specify that the system can move from one state to another, triggering the transition only if the associated constraint is satisfied. Several variants of timed automata have been proposed in the literature. We consider in our work timed automata with ε-transition (i.e., silent transitions). As an example, figure 1-6 depicts timed automata that is made of two states s_0 and s_1 and the clock variable x. this automaton includes a transition labeled Toilet from state s_0 to $s_1$which is guarded by the condition x >= 8h as well as ε-transition from state s_1 to $s_0$which is guarded by condition x == 24h.

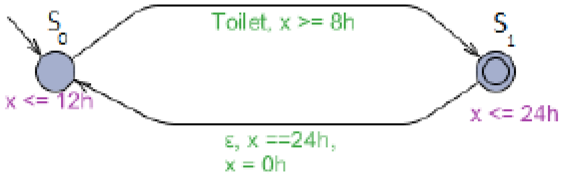


Figure 1-6 Example of a timed automaton

Definition 1.(Timed automata) A timed automata is a tuple $A = (S, s_0, \Sigma, X, Inv, T, F)$ where:

1) S is a finite set of locations or states of the automaton and $F \subseteq S$ is a set of final states;
2) $s_0 \subseteq S$ is a set of initial locations;
3) Σ is a finite set of transition labels;
4) X is a finite set of clocks;
5) $Inv: S \rightarrow \phi(X)$ associates an invariant to each state of the automaton;
6) $T \subseteq S \times \Sigma \cup \{\varepsilon\} \times \phi(X) \times 2^X \times S$ is a set of transiction. A transition $(s, a, \phi, \lambda, s')$ represents an edge from location s to location s' on symbol a. ϕ is a clock constraint, and the set $\lambda \subseteq X$ gives the clocks to the reset after firing such a transition.

Definition 2. (Time constraints) We define below the constraints allowed to express to invariants and guards [21]. The set of clocks is denoted by X. A clock constraint ϕ is defined by the grammar:

$$\phi := \mathrm{x} \preceq \mathrm{c} \mid \mathrm{x} \succeq \mathrm{c} \mid \mathrm{x} \prec \mathrm{c} \mid \mathrm{x} \succ \mathrm{c} \mid \phi 1 \wedge \phi 2 \tag{1-1}$$

Where x is a clock in X and c is an integer positive value. The set of clock constraints using clocks of X will be denoted by $\phi(X)$.

To formally define the semantics of timed automata we introduce the notion of variable valuation. We consider as a time domain the set of non-negative reals $R^{\geq 0}$. Let X be a set of variables with values in $R^{\geq 0}$. A (variable) valuation $v: X \rightarrow R^{\geq 0}$ is a mapping that assigns to each variable $x \in X$ a time value $v(x)$. Each state of the automata is represented by a pair (s, v) where s is a state of S and v is an evaluation over X s.t. v satisfies the invariant Inv(S). The semantics of a timed automaton A is then expressed in terms of two types of transitions [21, 35]: 1) Action transition: this type of transition can be triggered instantaneously if the current value of the guard is checked. Formally, for a state (s,v) and a real-valued time increment $d \geq 0,(s,v) \to^{d} (s,v+d)$; 2) Time transition: this type of transition consists of staying in the same state and increases the values of clocks respecting the invariant. Formally, for a state (s,v) and a transition (s,a,ϕ,λ,s') , v satisfies $\phi,(s,v) \to^{a} (s',v \mid \lambda : 0 \mid)$.

We use model checker UPPAAL to check home care plans. UPPAAL is a tool box for validation (via graphical simulation) and verfication (via automatic model-checking) of real-time systems, which is jjointly developed by Uppsala University and Aalborg University. The model-checker Uppaal is based on the theory of timed automata [21] and its modeling language offers additional features such as bounded integer variables and urgency. It consists of two main parts: a graphical user interface (GUI) (executed on the users work station) and a model-checker engine by default executed on the same computer as the user interface, but can also run on a more powerful server. The query language of UPPAAL, used to specify properties to be checked, is a subset of TCTL (timed computation tree logic). In UPPAAL, systems are modeled using timed-automata, which are finite state machines with clocks, where clocks are variables which can evaluate to a real number and which can be defined in each automaton in order to measure the time progress. All clocks evolve at the same pace in order to represent the global progress of time. The actual value of a clock can be either tested or reset (not assigned).

An UPPAAL model is built as a set of concurrent processes, which are graphically designed as a timed automaton. A timed-automaton is represented as a graph which has locations as nodes and edges as arcs between locations (see Figure 1-7). A guard is an expression which uses the variables and clocks of the model in order to indicate when the transition is enable, i.e., may be fired, where several edges may be enabled at a specific time but only one of them will be fired, so it leads to different potential interleaving.

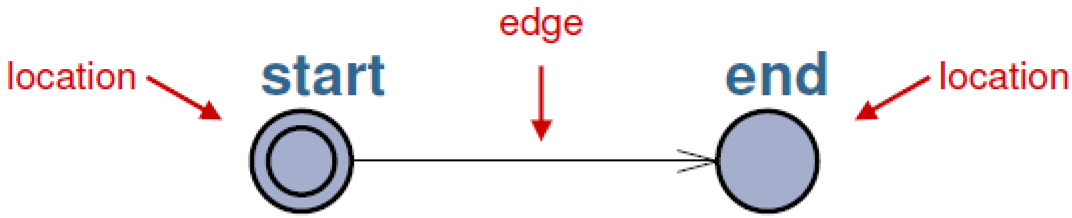


Figure 1-7 Example of an UPPAAL model

1.4 Main content and organization of the thesis

The thesis puts up with a novel DSL (domain specific language)-based approach to manage home care plans and developed a tool for visualization which is easy to express, monitor and advance the process of care management rapidly and exactly. Meanwhile, using UPPAAL to validate the care plans and providing the useful information to the end users (coordinators) who are responsible for specifying the home care plan for patients.

The thesis is presented in four parts. The first chapter gives an overview of the content of home care plan management related to current workflow-related technologies respectively. This chapter also addresses related work in the field, which has informed and guided the author towards his solutions. The chapter 2 explains the overall requirements of the whole project according to interview with the end users. The chapter 3 illustrates the overall system design for home care plan management and explains in detail the approach taken to provide a flexible goals-focused solution to meet the end users' needs. The chapter 4 describes how to implement the project step by step and addresses the practical work and evaluation by testing.

1.5 Brief Summary

This chapter illustrated the background of this topic and the purpose of this thesis. Then, the status of research in fields that related to this topic has been stated. Because using the traditional method home care plan which is unstructured process can not be managed effectively, we put forward this novel approach to manage it. Finally, the main content and the organization of thesis have been elaborated.

Chapter 2 System Requirement Analysis

2.1 The goal of the system

The goal of the thesis is to put up with a novel approach to manage home care plan and develop an intelligent software tool, which is capable of analyzing, monitoring and eventually giving the feedbacks to the interface. Meanwhile, as a part of Plas'O'Soins, I plan to design the use-friendly interfaces and robust database for other groups. The requirements from coordinators (end users) who are responsible for specifying home care plan are to be met with proper compensation. The novel approach is responsible for finding a solution if the requirement was met. If the requirements cannot be met, the tool decides to verify home care plan need to be made to optimize the design. The difficulty of this research project is in designing an effective method to resolve this problem.

2.2 The functional requirements

The whole structure of the novel approach is illustrated as figure 2-1.The approach proposed is based on the following processes: at first we have modeled the care plan management process using UML (class diagrams), and repeatedly revise the meta-model according to the requirements that always change. Then we have analyzed the relationship between classes and translate them to class object automatically. Then we configured the hibernate xml and translate them to rational database, which consist of database of project with the knowledgebase, while class object can obtain all the data from database, where rational database stores the data related to home care plans and knowledge base defines several default attributes to implement intelligent system. In the application level, a platform which is a RIA (Rich Internet Application) will be designed to simply the operation of home care plan coordinators to specify home care plan and gather information of patients and home care plan respectively. Each component on the platform will match the each class in the coding level. Finally because there are some timed constraints in care plan management, the timed arc of automata will be conducted by the algorithm we have designed regarding the information of home care plan and the verification of automata is to trace a new Business Process model associated to the improved process, verify the time constraints, elicit the new tool's requirements and define performance indicators in order to control the whole system.

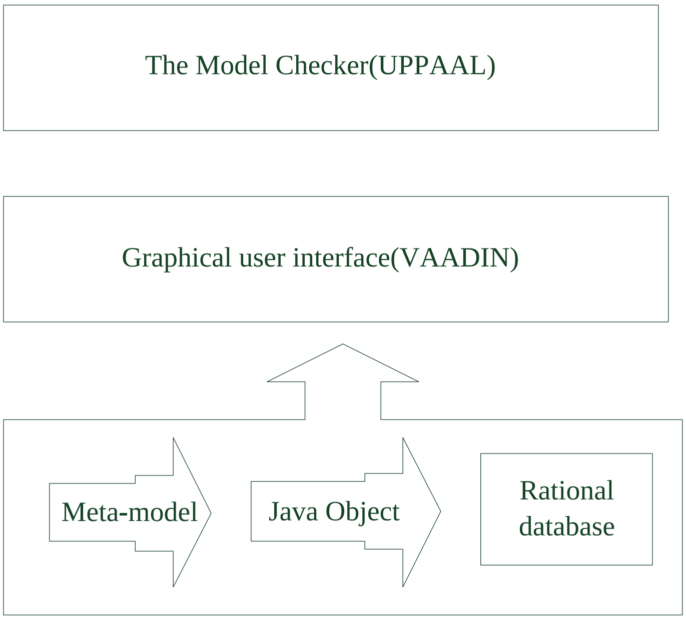


Figure 2-1 the structure of the novel approach

2.2.1 Translation from meta-model to rational database

Most of the software developers and industrial groups giving huge effort trying to place together both object oriented technology and relational model in order to fulfill the needs of software development process[10][11].Translation from model to rational database aims to convert abstract model to rational schema in order to manipulate data. It consists of two processes: translation from model to class object and establishment the relationship between class object and rational base. The meta-model consists of basic concepts of object-oriented concepts such as meta-class, metaproperty, operation and association, etc. It defines the pattern of all the syntax units and their composition relationship in the UML. The class diagram is the form of instantiation of meta-model. Based on the requirement of the project, the rules of meta-model was defined and meta-model will be represented in the class diagrams.

The requirements of this part are showed as follows:

- Using Rational Rose or other Unified Modeling Language (UML) software design tool to establish our model and modify it based on the requirements of home care plans. Our old model is shown in the figure 2-2. It consists of two core contents: Act and intervention. Act contains activities, time, duration, and the level of situation and information of nursing stuff specified by coordinators. Intervention aims to restore the information in different orders.

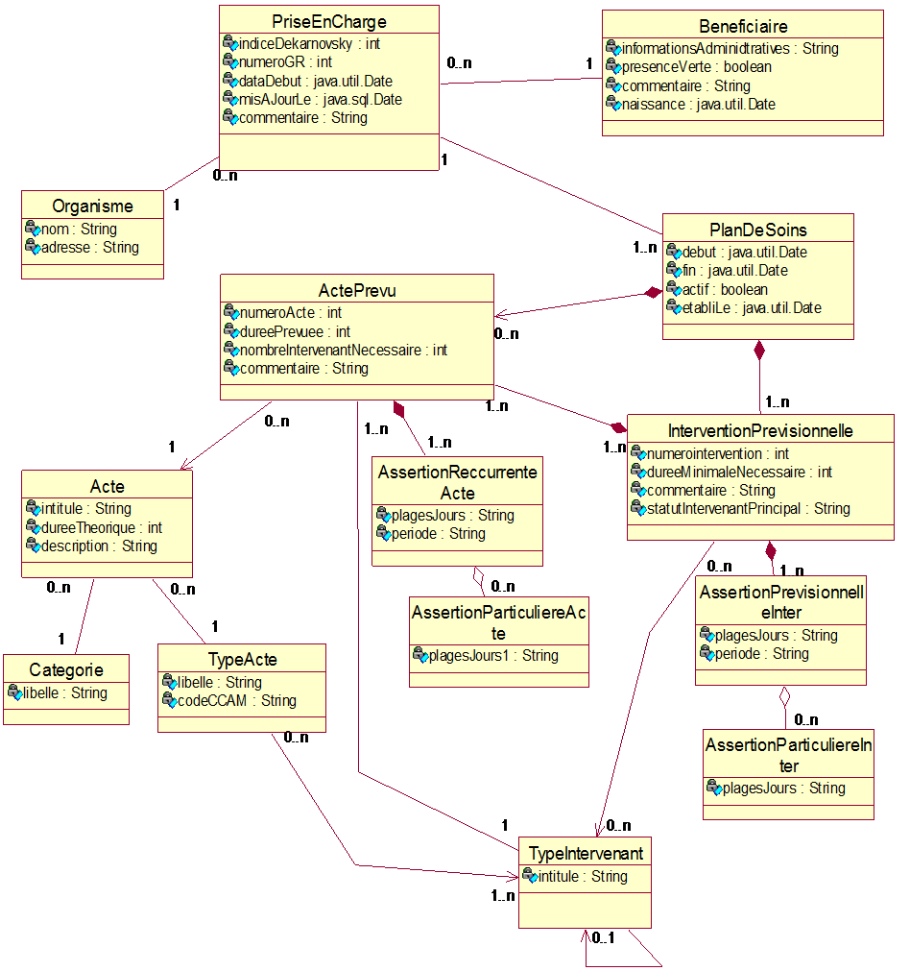


Figure 2-2 Meta-model(class diagram)

- Generate class object in java automatically and check syntax of each class diagram.
- Add the annotations or write mapping file on java project to generate the tables in the rational database.
- The rational database and the knowledge base make up of storage of our project. Logging data in the database manually.

- Establishing the relationship between class object and database, that is to say, keeping our model and class object synchronized with the database while both are being iteratively developed.
- Providing the interface to GUI to read accurate data from the database.

2.2.2 Graphical User Interface

The user interfaces should be business-server web application. They are defined to describe the whole process of specifying home care plans. The users of GUI are primary medical coordinators or nursing navigators. Doctor coordinators are in charge of specifying and monitoring the whole process and nursing navigators focus on checking the information whether they are accurate or not. The functional requirements are showed in Figure 0-3. As a coordinator, he could query the existing patient or create a new patient. For the existing home care plan, the coordinator could delete, modify and query it. Otherwise, the coordinator has to create a new home care plan for one patient. After the home care plan is finished specifying, the coordinator could classify home care plan according to his requirement. The detailed requirements are described in following subsections. This paper mainly focuses on querying the existing patient's home care plan, specifying the home care plan for the patient as well as classifying and analyzing the home care plans.

Especially, when user wants to use this system, you could choose the language (English or French) which is familiar for them at the beginning of running system. The end user has two options: 1) they can query the existing patient and related information and home care plan; 2) if the patient was not existed, they can create a piece of record of a new patient and specify the home care plan for him.

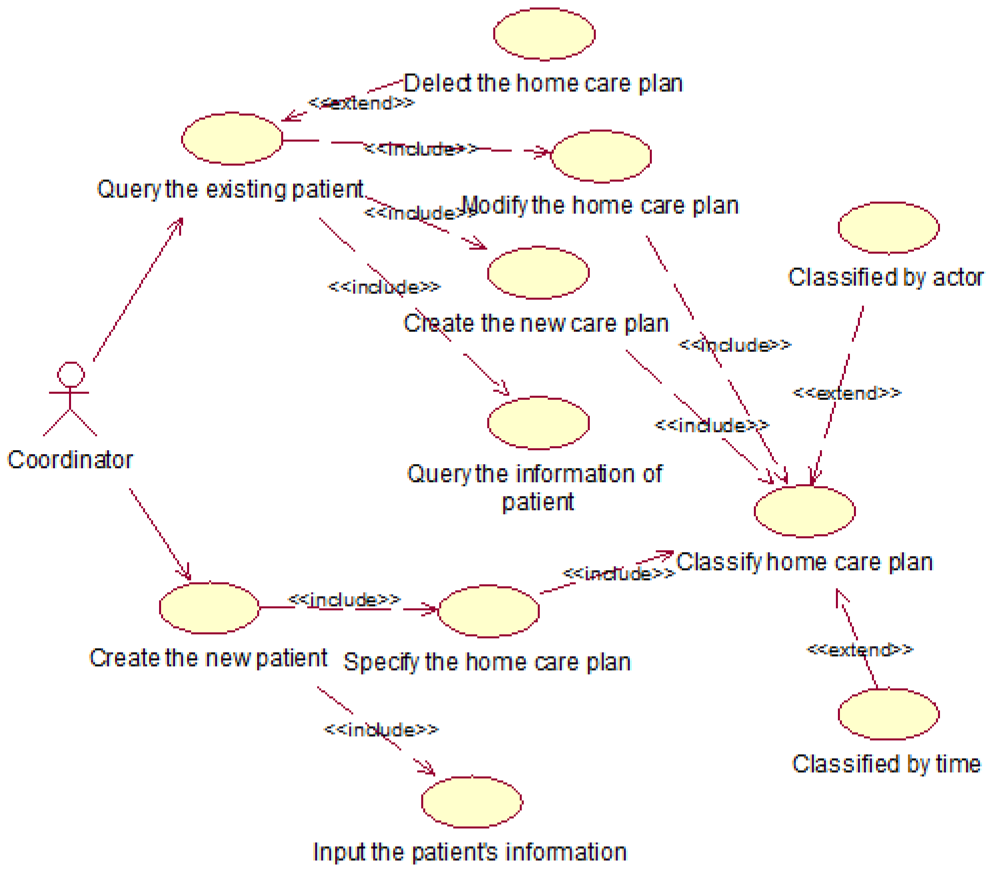


Figure 0-3 Use case diagram

1.1.1.1 2.2.2.1 Information of patient

Coordinators can query or create a piece of record of patient’s information. Patient’s information is divided into three parts: personal information, address information and modified information, where personal information includes name, age and sex … some basic information. Address information describes very detailed description of location of patient in order to care at home. For modified information, it shows that the person and date to modify or create patient’s information. Detailed patient’s information is shown as user case table (see Table 2-1).

Table 2-1 User case table of patient

Attribute Name	**Unit of measure/calculator**	**Example**
Name		Yuanling Wu
Date of birth		01/03/1988
Age	Calculated by date of birth	25
Sex	Mr./Ms./Mrs.	Mr.
Height	meter	1.72
Weight	kg	85
Body Mass Index	$BMI = {weight}/{height^2}$	28.73
Telephone number	Country code + number	0033064267777
Address	Street + city + post code + detailed building and apartment	
Number and name of street		38 Street Happy
Post code		33000
City		Clermont Ferrand(France)
Building		People Building
Apartment		Apartment 243
Zone	Provided by knowledge base	Northeast of CF
Modified person	Default value (coordinator)	Kahina
Modified date	Default value (current date)	08/05/2013

1.1.1.2 2.2.2.2 Specify home care plan

This thesis aims at proposing an approach based on a domain specific language. A domain specific language (DSL) is a language designed to express a solution to a problem within a specific domain [28]. A DSL can be tailored to a business or industry domain. While DSLs have already been used to some extent in the fields of computer science and mathematics [29], their use in the health care domain is less widespread. We describe below the main concepts of a DSL tailored to express home care plans. The proposed DSL provides high level abstractions that can be used by a care coordinator to design a care plan for a given patient.

The main building block in a care plan is the notion of activity. An activity denotes a medical or a social service provided to persons in their own homes. The proposed DSL includes several predefined activities identified by our analysis of the application domain.

Each activity is associated with a description which provides additional information about the activity. In particular, a description of an activity includes the required qualification of the actors that are allowed to carry out the activity as well as the temporal constraints that specify when the activity must take place. More precisely, a temporal constraint is expressed as a triplet (Period, Days, Time ranges), where:

1) ‘Period’ specifies the time period during which the activity is defined.
2) ‘Days’ indicates the days within a period in which an activity must take place.
3) ‘Time ranges’ indicates the time slots in which the activity can occur.

The formal expression of this triplet using the BNF (Backus-Naur Form) is shown as follows. Several triplets can be associated to a same activity in order to permit the specification of irregularities and exceptions. Triplet is the basic component of a general declarative language that we have proposed for specifying near regular repetition of activities [31].

Period

```
<period>::=<starting-date>”-”{<ending-date>}?
<starting-date>::=”today”|<date>
<ending-date>::=”today”|<date>
```

Days

```
<days>::={<date>+|{<day-of-week>}+|<specific-day>
Where <date>::=/a date with the format dd/mm/yy/
<day-of-week>::=”Monday”|”Tuesday”|”Wednesday”|”Thursday”|”Friday”
|”Saturday” |”Sunday”
```

```
<specific-day>::=”Odd-day”|”Even-day”|”Holiday”
Time ranges
<time-ranges>::=(<Integer>”times”)|{<Interval>}+”nothing”
Where <integer>::=/number of occurrences of the activity in the day/
<interval>::=<string-form>|<pair-form>|<hour-form>
<pair-form>::=”morning”|”noon”|”afternoon”|”evening”|”night”
<hour-form>::=<hour>/an hour with the format hh/mm/
```

Each patient could have different home care plans. For the existing patient, a primary medical coordinator can review his care home plan and take operations on it, such as modify and delete it. Meanwhile, for the new patient, a coordinator will specify home care plan after inputting the information of the patient. A home care plan concludes several activities, such e.g., medical, nursing, social needs which are a comprehensive assessment of patient's needs as well as its social and physical environment. Each time a coordinator can add one or more activities for a care plan. An activity corresponds to a health care or a supportive care which is associated with temporal constraints (e.g., an irregular frequency) and to a required qualification (a nurse, a nurse auxiliary, etc.). Table 2-2 shows a simple example of a content of a temporality for a given patient’s activity ‘Dress’.

Table 2-2 Excerpt of a temporality of Dress

Period	Time Intervals	Days
From 01/01/2013 To 03/31/2013	Morning	Everyday
		Monday-Friday
		Tuesday, at 08 a
	Evening	Every day except Sunday
		Festival

This example illustrates some important concepts of a care plan. The first column period shows the period in which this plan is valid. For each activity, the following information are given: (i) temporal constraints associated with the activity and expressed in terms of a Time slot (e.g., morning, afternoon or evening) and Repetition (e.g., every day except Sunday), and (ii) the required qualification to carry out the activity (e.g., nurse auxiliary, nurse). Irregular activities are inherent to care plans. For example, a given activity, e.g., "toilet", may be associated with complex temporal constraints, e.g., every two days except Sunday evening. Activities are then grouped together within interventions which are then performed in a rotating schedule in accordance with the specifications of the care plan.

1.1.1.3 2.2.2.3 Classify home care plan

For the specified home care plan, coordinators could generate classified table by actors, activity and different time. It has the same information as the home care plan but different structure of classifying care plans. Table 2-3 shows a simple example of a content of a home care plan.

Table 2-3 Excerpt of a home care plan

Activity name	**Period**	**Time Range**	**Days**
A1	01/01/2013-31/03/2013	Moring, Evening	Monday-Saturday
	01/01/2013-31/03/2013	Morning	Sunday
A2	01/01/2013-31/03/2013	Evening	Monday-Sunday
	01/01/2013-31/03/2013	08H00	Monday-Wednesday

The coordinator can classify home care plan according to his needs. For instant, the above home care plan will be classified based on different time range, different activity name or type, different days’ repetitions. It will generate another schema to present the home care plan, which help coordinators find the feasibility of the specified home care plan.

In addition to the notion of activity, another important concept of the DSL lies in the notion of Intervention. An intervention is a collection of activities that can be scheduled together. Interventions are defined by grouping together the activities that can be performed by a same professional and which occur in the same time range. Interventions may be specified manually by the coordinator or computed automatically from the specifications of the activities and then proposed to the coordinator for validation.

2.2.3 Model monitoring and detection module

When all the information of home care plan was collected, we will analyze the home care plan and pick up the useful information to generate timed-automata (singular: automaton). A two steps approach which works as follows: (i) first, basic activities of a care plan are translated into timed automata using an appropriate algorithm, then (ii)a global care plan is generated by composition of the components. Each activity a of a care plan is mapped into a timed automaton. As the input of automata, the detailed specification of a basic activity: ({Activity name (e.g. Toilet). Days (e.g. every day), Time range (e.g. Morning), Period (01/01/2013-09/08/2013)}.

Formal verification of care plans is a crucial problem. To enable automatic verification, it is then essential to map the DSL concepts into a formal model. In our approach we use timed automata [22] to formally describe care plan constructed using the DSL and then we rely on the large body of theoretical results and existing implemented systems for this class of automata in order to support verification and monitoring of care plans. The requirement of this module:

1) As the input of the automata, a set {activity name, days, time range, period} will be proposed.

2) Using the UPPAAL Model checker to generate the network of automata and verify it.

3) Through the analysis of UPPAAL Model checker, it will monitor and give feedback to the graphical user interface.

4) Design and implement an effective algorithm in order to generate the corresponding timed automata.

5) Generation automatically the whole care plan from the basic activities. Interestingly, the care plan can be also described by means of a timed automaton which is obtained by composition of automata representing basic activities that have been generated in the first step. The composition is achieved using the asynchronous product of activities automata

2.3 The un-functional requirements

The nonfunctional requirements are showed as follows:

- **Reliability**: The approach should ensure that home care plans management effectively and simplify coordinators’ operations.
- **Automaticity**: The approach should make the care plan management process run automatically beginning with the existing model.
- **Real-time ability**: The approach should minimize the time used to generate home care plan and give the feedback as soon as possible.
- **Expendability**: if requirements of home care plans is changed, that is to say, the new models or the new information in the knowledgebase are added or the original ones are cut, the whole process will not be modified.
- **Code readability**: All Classes, methods, and important properties have to be commented. All Commented fields have to be spell checked. The code of software should be readable and easily understandable to future engineers that continue this project.
- **Intelligent**. This requirement refers to the intelligent association when the primary medical coordinator input the information. Because there is knowledge base in the database, normally the system will provide the prompt to the end user.

2.4 Brief summary

In this chapter, the project, which comes up with a novel to manage home care plan, its requirement has been stated and analyzed. From requirement type point of view, the project requirement can be classified as two types: functional requirement and non-functional requirement. In the functional requirement, it can be also divided into three parts: transformation from meta-model to rational database, graphical user interface module and model monitoring and detection module.

In functional requirement of transformation from meta-model to rational database, the modified meta-model will be modified. Also the primary process was fixed: firstly the meta-model in form of class diagram was transformed to java code automatically, and using ORM technique to generate rational database.

In functional requirement of graphical user interface module, the information of patient will be described in the detail; also, the specifying process of home care plan was proposed and the graphical user interface was shown because of prototyping development.

In functional requirement of model monitoring and detection module, a formal framework based on timed automata will be put forward. The whole process of this module is introduced briefly and the requirement of the process is listed in detail. We need design an algorithm using the information from database to generate automata. Using the UPPAAL Model checker, the automata will be monitored and verified. The final purpose is to generate a piece of home care plan by modified automata.

In non-functional requirement of mapping mechanism, there were totally five requirements: reliability, automaticity, real-time ability, expendability, code readability and Intelligent.

Chapter 3 System Design

According to the requirement analysis, the design phase of this project can be divided into three parts. First of all the meta-model will be modified in form of the class diagrams and be transformed to database and knowledge base. In this part, the meta-model transformation approach should be firstly established in order to establish an accurate database to restore the information of patient and related home care plan. Secondly, the graphical user interface could be proposed for coordinate to collect and inquire the patient information and specify home care plans for each patient intelligently. During this part, a modified V-model cycle technique is used. In the final part, using the data from database, automata will be generated automatically to analysis and monitor home care plans and detect the user interface intelligently.

3.1 A domain specific language for home care plans

A domain specific language (DSL) is a language designed to solve a problem within a specific domain. The proposed DSL provides high level abstractions that can be used by a care coordinator to design a care plan for a given patient. We describe below the main concepts of a DSL tailored to express home care plans.

The first important concept in a home care plan is the notion of activity. An activity denotes a medical or a social service provided to persons in their own homes. The proposed DSL includes several predefined activities identified by our analysis of the application domain. Examples of predefined activities:

1) Health services: monitor medications, drug injection, aftercare, etc.

2) Activities of daily living: bathing, assist with meal planning and preparation, dressing, maintain clean household, etc.

Another main concept is the notion of Intervention. An intervention is a collection of activities that can be scheduled together. Interventions are defined by grouping together the activities that can be performed by a same professional and which occur in the same time range. Interventions may be specified manually by the coordinator or computed automatically from the specifications of the activities and then proposed to the coordinator for validation.

Besides the aforementioned concepts, the proposed DSL is enriched with additional constraints derived from the domain knowledge (e.g., medical knowledge represented in ontologies, etc.).

3.2 Design of the system architecture

Figure 3-1 describes the whole system architecture of the novel approach. There are three layers: database layer, graphical user interface layer and logic layer. For database layer, combining with the existing meta-model and the requirements by coordinator, all the new meta-model will be modified in the form of class diagrams. After that, all the meta-model should be transformed to java codes automatically using tools and finally be transformed to database and knowledge base using object-relational mapping technique. After establishing the database, a graphical user interface will be proposed for coordinators who are in charge of making home care plans to collect patients' information and specific make care plans for each person, which is proposed by using Vaadin framework. For the logical layer, database provides information, especially temporality (periods, days, time ranges) to generate the timed automata using model checker tool. It is worth noticing that the algorithm which will be proposed by my teams analyzes feasibility of the home care plan and monitor the home care plan specifying process.

3.3 Design of database layer

In the database layer, the whole process is illustrated as figure 3-2. Other than the traditional construction of database, the whole proposed transformation process uses the meta-model in form of class diagrams to describe the basic requirement of project in order to create the relationship between meta-model and rational database. Firstly, meta-model should be revised and represented in form of class diagrams and then translate it to java codes using Rational Rose. Each java class corresponds to the component in the graphical user interface. And then through ORM (hibernate), rational database and knowledge base and the connection between java class and database will be established. Two kinds of supported technologies will be integrated together and the proper method is adopted to generate corresponding database.

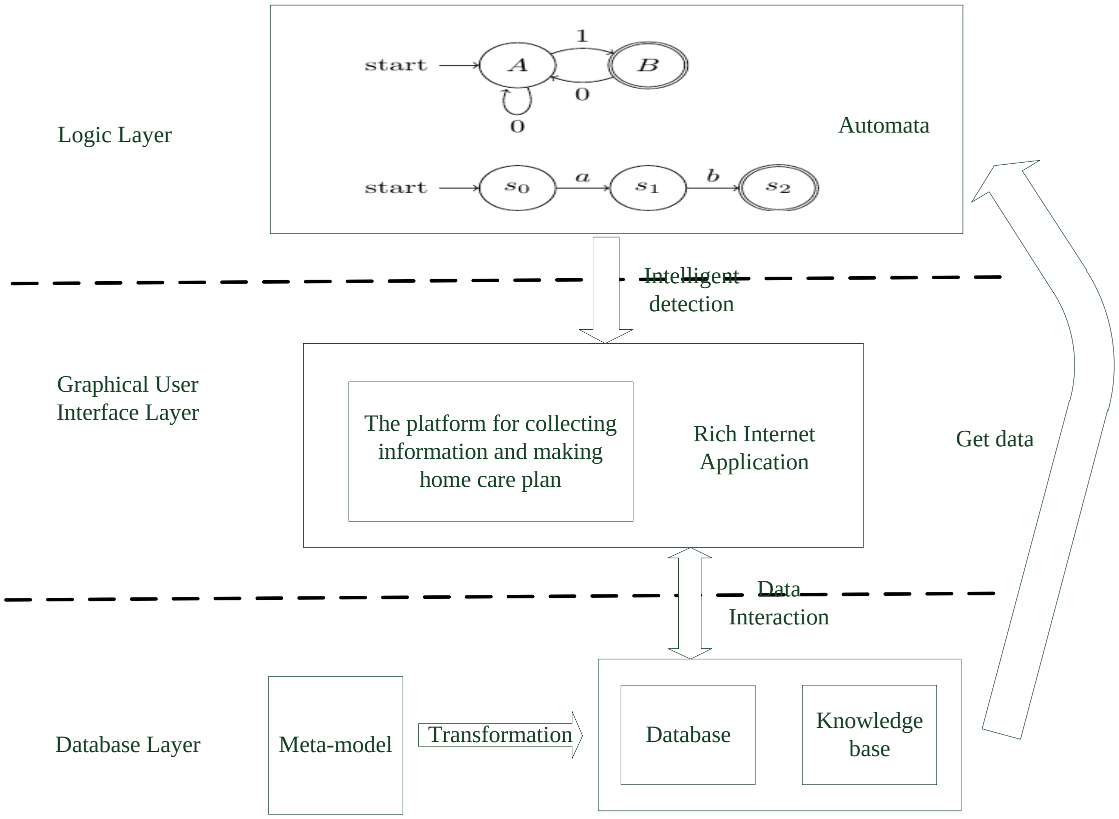
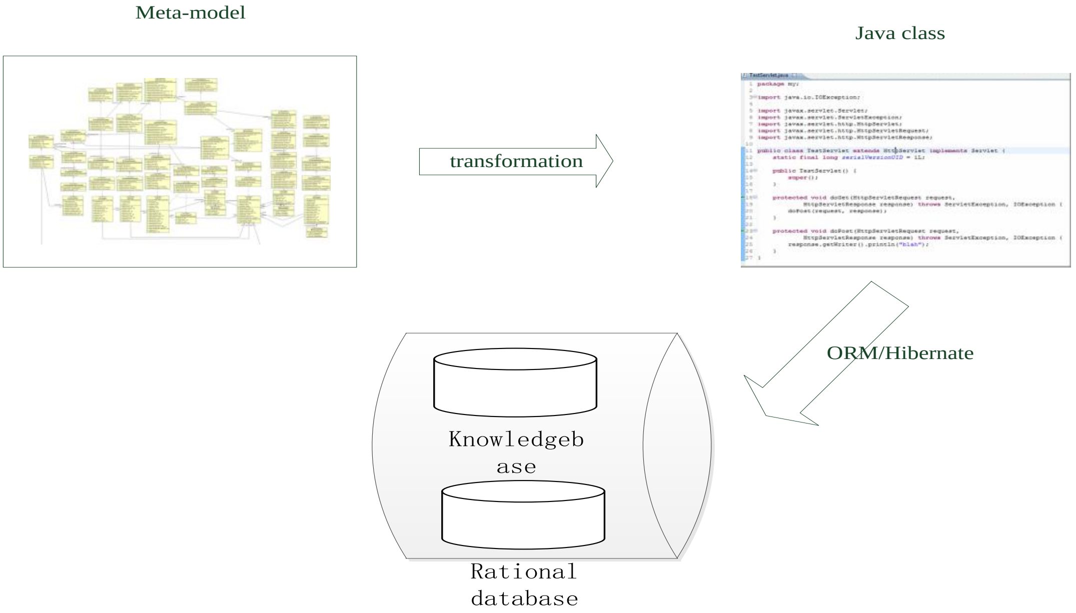


Figure 3-1 System architecture diagram



```
package my;

import java.io.IOException;

import javax.servlet.Servlet;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class TestServlet extends HttpServlet implements Servlet {
    static final long serialVersionUID = 1L;

    public TestServlet() {
        super();
    }

    protected void doGet(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request,
            HttpServletResponse response) throws ServletException, IOException {
        response.getWriter().println("blah");
    }
}
```

Figure 3-2 Model transformation methodology process in database layer

3.3.1 Design of the meta-model

As stated in requirement analysis section, the meta-model was described in the form of class diagram. They could be classified in five packages: projected care plan, patient, activity, actors, supported. The description of each package is as follows:

1) Package projected care plan. The projected care plan is the core package of the meta-model. The Class Beneficiare stands for patient, which contains the all the information of patients and each patient has many different care plans, which is describe in the Class PlanDesoins. The Class ActePrevu records each activity for each patient and each activity has different temporality which contains periods, days and time ranges as input of the generation of automata. As shown in Figure 3-3, the Class InterventionPrevisionelle will be classified by the Class ActePrevu. All the class consists of the whole home care plan.

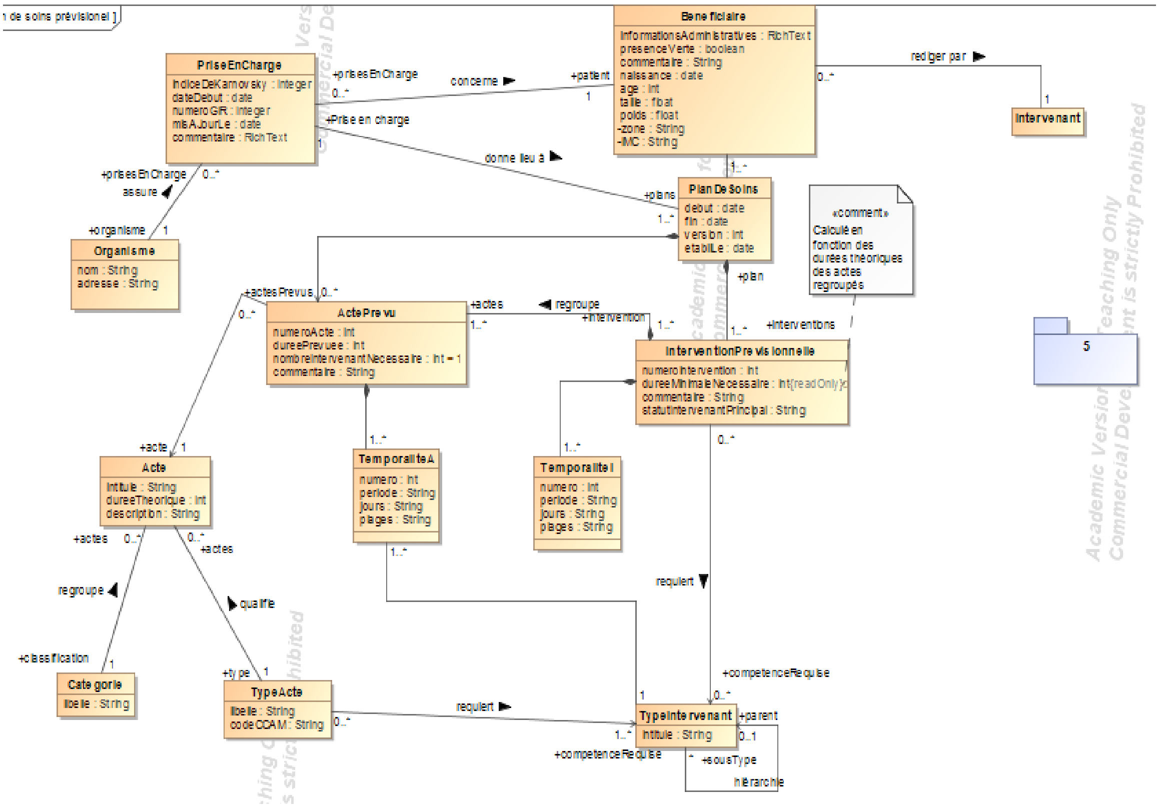


Figure 3-3 Package Projected care plan

2) Package Patient (see Figure 3-4). Beneficiaire (patient in English) class inherits the Personnel class which records the basic information of persons. Patient

class records the information of patient. For class Personnel, there are two sub-classes: patient class and actor class. Thus, each Personnel class has its temporality. For patient class, class Temporilate stands for the temporality of an activity in the home care plan. Each patient class assists the patient class.

3) Package Acte (see Figure 3-5). Class Acte records information of each activity (name, default duration, description). Type of activity has multiple activities, so they are the many-to-one association relationship. A type of activity corresponds to different type of actors. Acte class is classified by the category which consists of the content of category.

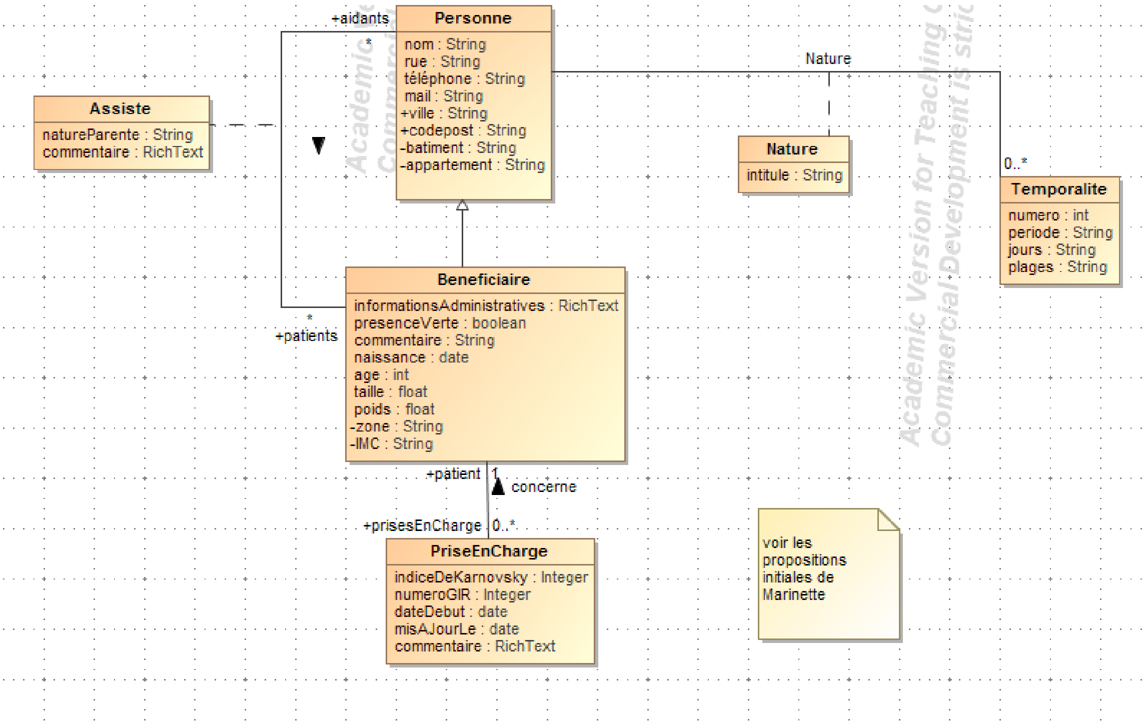


Figure 3-4 Package Patient

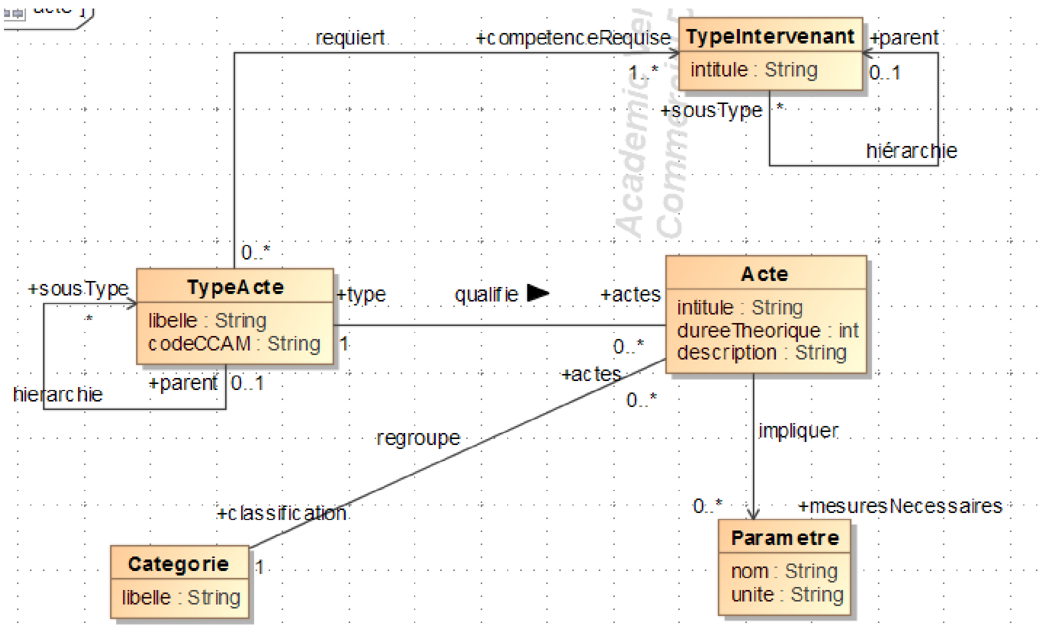


Figure 3-5 Package Activity

4) Package Intervenant (see Figure 3-6). Class Intervenant records information of each actor who could be nursing staff or coordinators, which inherits Personne class. Each actor has his organization recorded in Organisme Class. Each organization corresponds to unique type of organization.

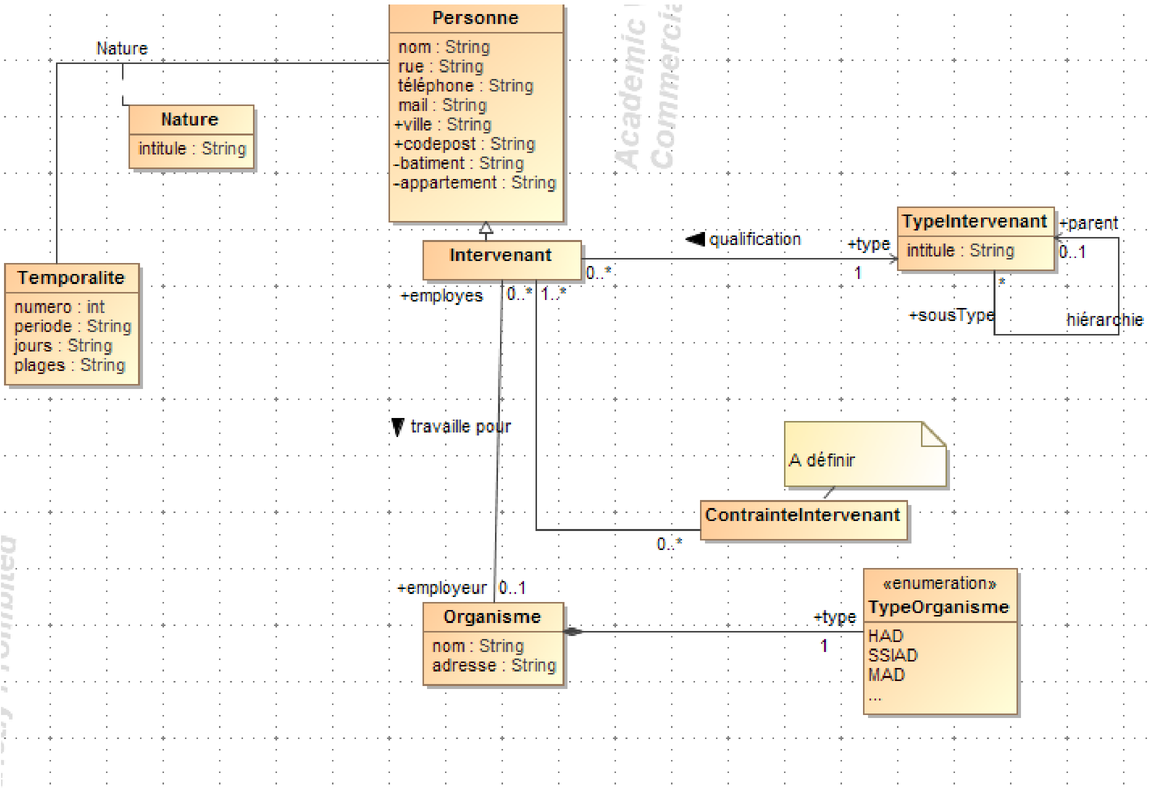


Figure 3-6 Package Actor

5) Package Support (see Figure 3-7). Each care plan will be in the document class which contains content, creating date and URL. Each patient has multiple support plans and one support has multiple home care plans. The support plan was established by actor class.

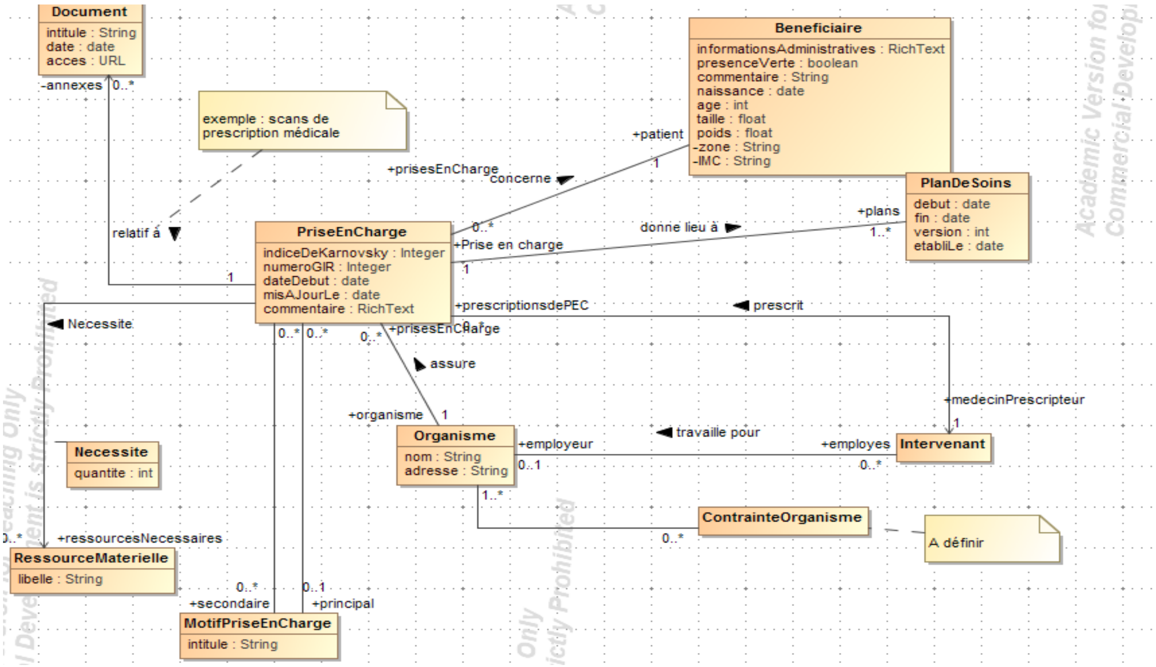


Figure 3-7 Package Support

According to the requirement, we use one UML modeling tool Rational Rose to create UML and connecting the connection between java code and rational database. Rational Rose is an object-oriented Unified Modeling Language (UML) software design tool for visual modeling which is the process of graphically depicting the system to be developed. Meanwhile, Rational Rose consists of a lot of tools for reverse engineering as well as forward engineering of classes and component architectures. In our project, we use code generation because code generation is limited to class diagram only and our meta-model will be described in form of class diagrams.

3.3.2 Transformation between java codes and database

Through transformation from java classes to rational database, sixteen key tables are created. All the information about patient's home care plan will be restored in the rational database. A part of them will be used for knowledge base.

The whole process is illustrated in figure 3-8. The class diagram will be generated to be java code by the rational rose automatically and the connection will be created through object/rational mapping. For the first step, because the class diagrams have been built up, we only need set the configuration of rational rose to generate java codes. For the object/rational mapping, we choose the Hibernate to create the connection between java code and rational database. Hibernate is used convert object data in JAVA to relational database tables. Its advantage is: 1) Hibernate automatically generates the SQL queries. 2) Hibernate provides data query and retrieval facilities and can significantly reduce development time otherwise spent with manual data handling in SQL and JDBC. 3) Makes an application portable to all SQL databases.

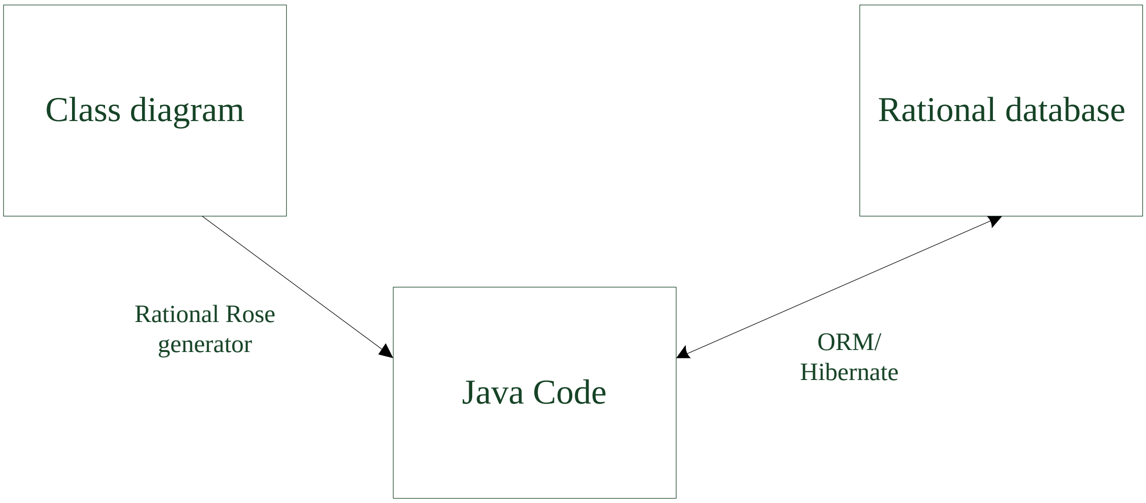


Figure 3-8 The whole process of transformation between java codes and database

Through the whole process, the rational database will be created finally. There are several tables in the database and some of them will be knowledge base. Acte table (see Table 3-1) contains all the activities' information, for example, name, type, duration of each activity, and description. Acte table will be a part of knowledge base to support associated information for the graphical user interface.

Table 3-1 Acte Table

Attribute Name	Key Type	Type	Description
id	Primary	int(11)	The ID of activity
intitule		Varchar(255)	The name of activity
dureeTheorique		int(11)	The duration of activity
description		Varchar(255)	The description of activity
categorieid	Foreign	int(11)	The ID of Categorie table
typeacteid	Foreign	int(11)	The ID of TypeActe table

TypeActe table (see Table 3-2) and table Categorie (see Table 3-3) table store the associated information of the activity including the type of activity, the category of each activity the etc. Each activity can have multiple types of activity and categories. They are also parts of knowledge base.

Table 3-2 TypeActe Table

Attribute Name	Key Type	Type	Description
typeacteid	Primary	int(11)	The ID of type of activity
libelle		Varchar(255)	The content of type of activity
codeCCAM		int(11)	

Table 3-3 Categorie Table

Attribute Name	Key Type	Type	Description
id	Primary	int(11)	The ID of category
libelle		Varchar(255)	The content of category

Beneficiaire table (see Table 3-4) contains all the information of patient, such as name, age, the date of birth, telephone etc. For the address of patient, it was described in detail: the street, the city, the zone, the name and number of building and apartment. For IMC (BMI, Body Mass Index in English), it will be calculated by the weight and the height of the patient. For recording who modifies or specify the home care plan for this patient, each Beneficaire table has multiple actors (Intervenant table).

Table 3-4 Beneficiaire table

Attribute Name	Key Type	Type	Description
id	Primary	int(11)	The ID of patient
Information Adminidtratives		varchar(255)	The information of patient
presenceVerte		int(11)	The Identity of patient
naissance		Datetime	The date of birth of patient

Table 3-4 Beneficiaire table (continued)

Attribute Name	Key Type	Type	Description
commentaire		varchar(255)	The description of the patient
age		int(11)	The age of patient
taille		double	The height of patient
poids		double	The weight of patient
rue		varchar(255)	The street of patient
ville		varchar(255)	The city of patient
zone		varchar(255)	The zone of patient
IMC		varchar(255)	BMI, Body Mass Index
batiment		varchar(255)	The building of patient
appartement		varchar(255)	The apartment of patient
nom		varchar(255)	The name of patient
telephone		varchar(255)	The telephone of patient
mail		varchar(255)	The email of patient
intervenantid	Foreign	int(11)	The id of actor who modify the patient’s care plan

PlanDeSoins table (see Table 3-5) contains the information of home care plan, such as the start time, the end time, the version of home care plan, modified date of home care plan. Each time the newest version of home care plan for patient will be received by coordinator. Each patient has multiple home care plans.

Table 3-5 PlanDeSoins table

Attribute Name	Key Type	Type	Description
id	Primary	int(11)	The ID of home care plan
debut		Datetime	The start date of home care plan
fin		Datetime	The end date of home care plan
version		int(11)	The version of home care plan
estabfiLe		Datetime	The modified date of

			home care plan
priseenchargeid	Foreign	int(11)	The ID of PriseEnCharge table
beneficiaireid	Foreign	int(11)	The ID of Beneficiaire table

Acteprevu table (see Table 3-6) includes the information of projected activity. Every care plan has many activities, such as medical, nursing, social needs. An activity has his number that starts with 0, the predicted duration and the number of actors who are nurses or professional carers.

Table 3-6 Acteprevu table

Attribute Name	Key Type	Type	Description
id	Primary	int(11)	The ID of projected activity
numeroActe		int(11)	The number of projected activity
dureePrevuee		int(11)	The projected duration of activity
nombreintervenantNecessaire		int(11)	The number of actors who are responsible for this activity
commentaire		varchar(255)	The description of activity
acteid	Foreign	int(11)	The ID of Acte table
plandesoinsid	Foreign	int(11)	The ID of PlanDeSoins Table

Temporalite table (see Table 3-7) includes the information of activity's temporality. Since it is the input of the automata, it is very vital for home care plan. Each activity has many temporalities, so the number is used to tell from them. The three key attributes (period, repetition, time intervals) are designed as Table 3-8.

In the table 3-8, the example of temporalite will be given. For an activity, it contains the act name. Repetition contains four situations: “Every day”, Specific days as a list of days (e.g. “Sunday, Monday”), detailed days as a list of days (eg. “07/03/13, 09/03/13”) and “Holidays”. Time ranges contain the values: “Any time”, “Morning”,”Evening”,”Noon”,”Afternoon”,”Evening”,”Night”, specific accurate time (“9H00”), “None” (“Act is not performed”). Period contains two date: start period date and end period date.

Table 3-7 Temporalite table

Attribute Name	Key Type	Type	Description
id	Primary	int(11)	The ID of temporality
numero		int(11)	The number of temporality
periode		varchar(255)	The period of temporality
jours		varchar(255)	The repetition of temporality
plages		varchar(255)	The time interval of temporality
typeintervenantid	Foreign	int(11)	The ID of TypeIntervenant table
acteprevuid	Foreign	int(11)	The ID of Acte Table

Table 3-8 Temporalite example

Attribute Name	Example
Activity	Toilet
period	01/01/2013 – 03/31/2013
repetition	Every day
	Monday - Friday
	Tuesday, at 08am
Time intervals	Morning

3.4 Design of graphical user interface layer

The user interface layer is responsible for describing the whole process of specifying home care plans graphically. Because it will be used for French local hospital, this graphical user interface will be designed and implemented in English/French. The whole process of graphical user interface layer is shown in figure 3-9. Firstly, the user could choose the language that they are familiar with. By searching the patient, they will get the information from the database. If the patient does not exist, the user can create a new record for this person. Otherwise, the patient could take operations on the information of patient, such as modify, delete and so on. Then after the information of patient does exist, the end user could specify home care plan for patient. If home care plan is not existed, the primary medical coordinator will create a new home care plan for this patient.

Specifying the home care plan is one of the most important tasks of the whole process. One home care plan consists of a set of activities which are made up of activity table. In the activity table, each row stands for description of activity. Figure 3-10 shows that the whole process of specifying home care plan. The process can be divided into four steps. 1) Choosing the activity. The system will propose some default activity names to the end user and meanwhile, the primary medical coordinator could input the activity name as he wants. 2) Entering the temporality for each activity. The temporality consists of three attributes: 1. 'Period' specifies the time period during which the activity is defined. 'Period' will be proposed by the system and it will have a range with the starting date and the finishing date; 2. 'Days' indicates the days within a period in which an activity must take place; 3. 'Time ranges' indicates the time slots in which the activity can occur. 3) Operations on the temporality table. When the primary medical coordinator finishes specifying the temporality, they can modify and delete the temporality in the temporality table. 4) Saving the temporality to the activity table. After modifying the temporality, it will be displayed in the activity table in the form of sub-table. 5) Generate activity table. Activity table consists of the temporality table and other default values proposed by knowledge base. The user could take operations on activity table and save all home care plans into database.

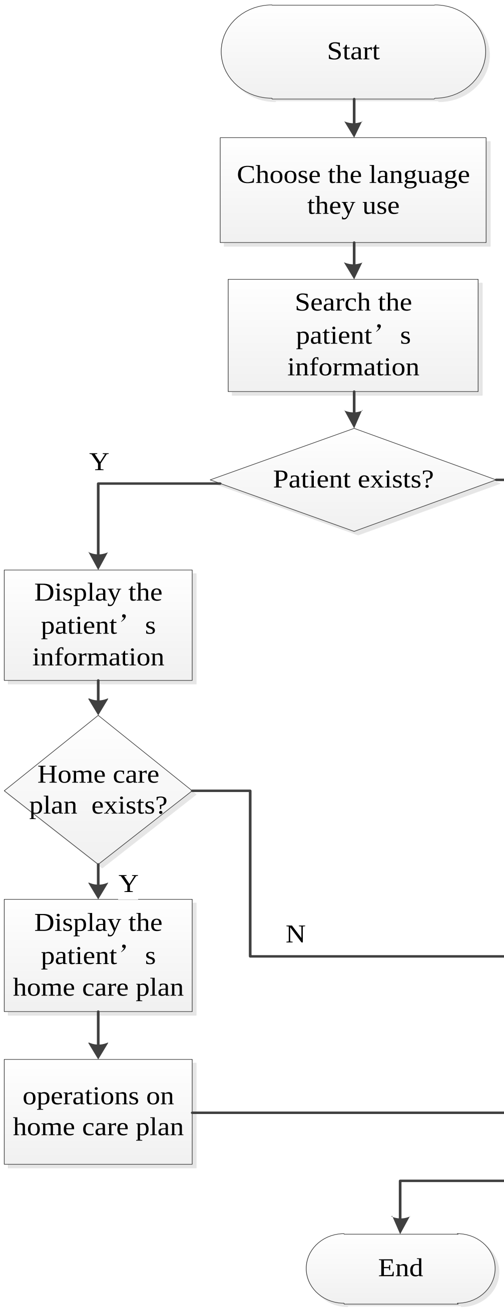

Start
Choose the language they use
Search the patient's information
Patient exists?
Y
N
Display the patient's information
Create the patient's information
Home care plan exists?
Specify the home care plan for patient
Y
N
Display the patient's home care plan
Create the home care plan
operations on home care plan
Save the home care plan
End

Figure 3-9 Flow chart of user interface layer

3.4.1 The architecture of the UI layer

The whole structure of the user interface layer is adopted by MVC (Model–view–controller) [6], which is a software architecture pattern which separates the representation of information from the user's interaction with it.

A controller can send commands to its associated view to change the view's presentation of the model (e.g., by scrolling through a document). It can also send commands to the model to update the model's state (e.g., editing a document).In our project, we put all the operation in the Listener package. When the view wants to take some actions, it will manipulate the classes in the Listener package.

A model notifies its associated views and controllers when there has been a change in its state. This notification allows the views to produce updated output, and the controllers to change the available set of commands.

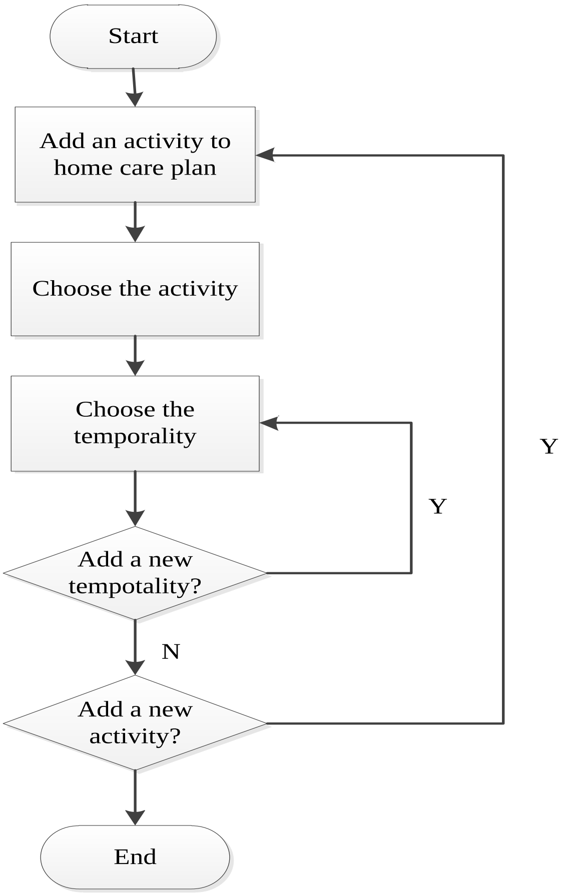


Figure 3-10 Flow chart of specifying home care plans

A passive implementation of MVC omits these notifications, because the application does not require them or the software platform does not support them [39]. We put TableModel and TreeModel class in the Model package, will be used for the update and change the view of the graphical user interface layer.

A view requests from the model the information that it needs to generate an output representation to the user. For the Vaadin framework, all the main pages will be written in the CarePlanManagerUI class and the custom component are written in the Vaadin composite which is a visual programming tool. All of UI design class will be replaced in the View Page.

If the primary medical coordinator asks the system to look for the patient who need specifying home plan, firstly, the CarePlanManagerUI object will call the verify() method . The verify() function checks in the database if the patient exists, and then patientInformationManager class will call the inform() method to send state of patient. There are three kinds of state: if state equals 0, it means that patient does not exist; if state equals 1, it means that patient exists but he did not have home care plan; if state equals 2, it means that patient and his home care plan does exist. If those two conditions are satisfied, the coordinator can review all the information of patient. If the patient does not exit, the CarePlanManager class will call record() method to input the patient information and store them into the database. Then PatientInformationManager class call the specify() function to assign the home care plan for this patient. We assume that there are 3 activities in the home care plan and each activity has 2 pieces of temporality. So we will use 6 threads to add temporality by add() function. After that, the activity table was established and coordinator can modify the home care plan, especially the information provided by knowledge database. The intervention table will be generated by generate() and will generate the activity table in the revise. If the patient has information in the database but did not have home care plan, it will be specified by CarePlanManager class directly. The process is shown in Figure 0-1.

3.4.2 Design of prototype of GUI

According to the requirement, we choose prototyping model to develop the graphical user interface. Here, a prototype is made first so that the final product is developed based on. A prototype is a model or a program which is not based on strict planning, but is an early approximation of the final product or software system. A prototype acts as a sample to test the process. In this work, for implementation, we finally choose Vaadin[4] framework, which is an open-source framework for developing Java Servlet based web applications as if they are Java desktop applications. For client side features, Vaadin framework extensively relies on the facilities of Google Web Toolkit7 (GWT). GWT is an open-source development system that allows the developer to write applications in Java and then compile the source code to JavaScript which can be run on all browsers. When using Vaadin, the developer can implement server side application using the full power and flexibility of Java and Vaadin automatically takes care of the client side.

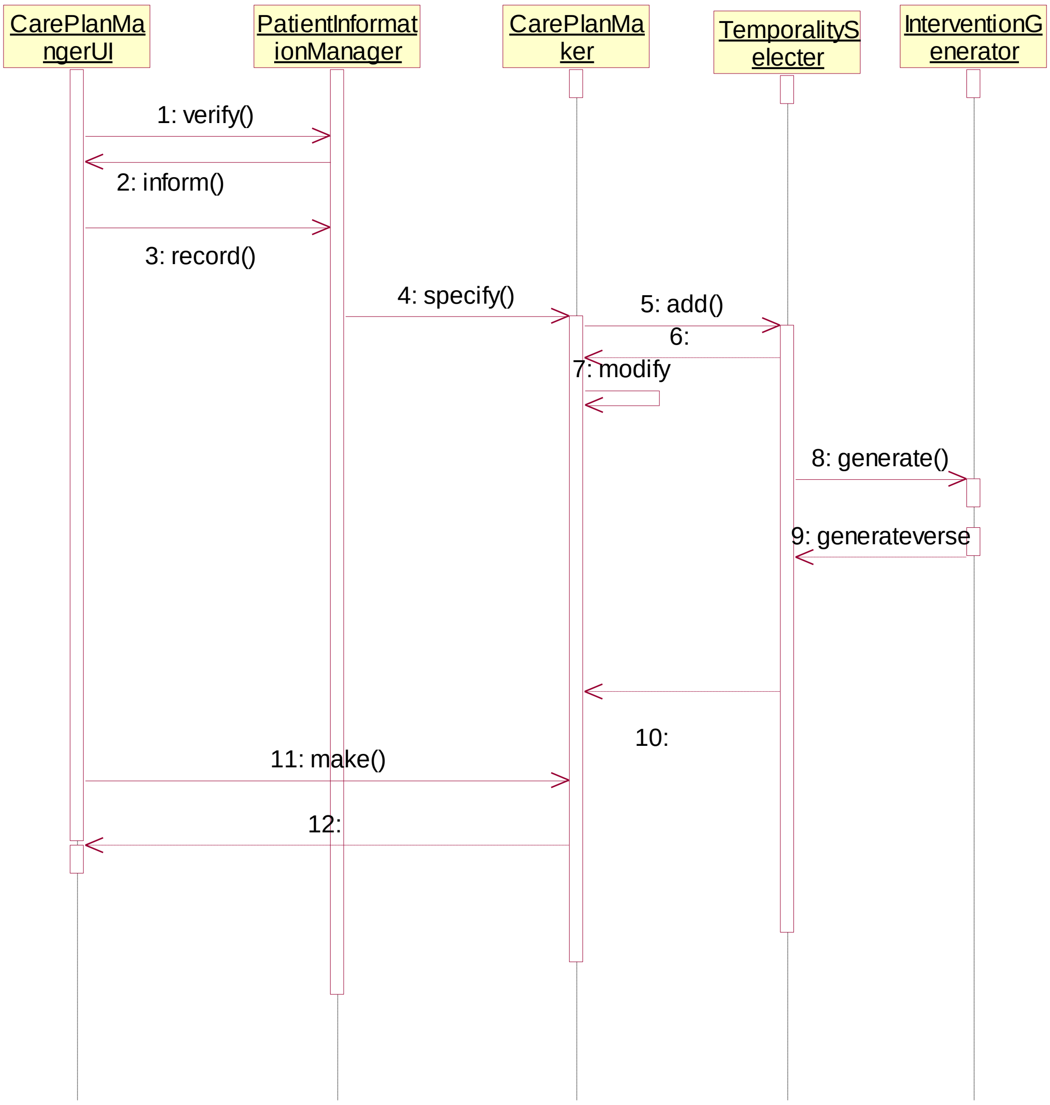


Figure 0-11 Specifying home care plan sequence diagram

When using web servers like Apache Tomcat, all the served web applications are located on the same Java virtual machine. However, Java lacks many important supportive features to run multiple applications on a single Java virtual machine. To patch this up, a widely adapted OSGi10 specification [17] has been published. OSGi introduces the dynamic component model which allows, e.g., updating, installing and removing components on-the-fly in a single Java virtual machine. Vaadin framework is compatible with OSGi. As the framework can be configured as an OSGi bundle, it is possible for other OSGi bundles to use it directly. In this way, Vaadin web applications can use the Vaadin framework bundle like a dynamic library resulting in very lightweight applications. Thus, the typical size of the Vaadin application bundle file in OSGi is less than 100 kilobytes, and the starting up of an application is a light and fast operation.

The input interfaces provide a set of information schema to input home care plan, which includes the information of patients, period of care plan, acts of care plan and allocated nursing staff. Each component on the input interfaces corresponds to each class in the class object. The output interfaces comprise a condensed planned intervention interface and an agenda care plan interface. They are generated automatically according to the description input by coordinators. In this part, prototyping development will be adopted since Prototyping model (see figure 3-1) could be used in software development as a visual aid to complement software requirements specification. The user interface is mainly divided into 3 pieces, patient information module, making home care plan module and classified care plan module.

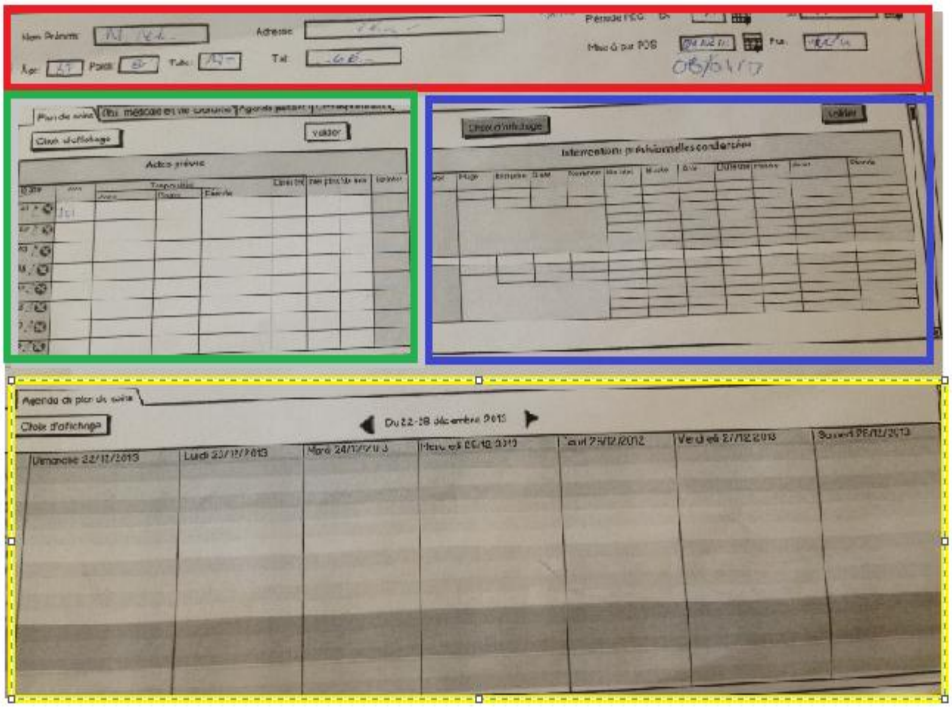


Figure 3-12 prototype of GUI

- Patient information module which is shown in the red pane is a panel which is composed of information of the patient. The primary medical coordinator records the information of patient in advance or querys the existing information of the patient in this panel.
- Making home care plan module is described in the tabsheet (see Figure 3-13). There are four main labels: Plan de Soins (care plan), Obs.medicale et vie cournate (Observation), Agenda patient(Agenda of patient) and correspondants (Correspondents). For the our approach, we focus on the first label which aims to making home care plan. The main function of this module is to fill in the activity table which is responsible for specifying different activities for each patient. For each activity, the ID of activity will be generated automatically from number 1. Activity name will be proposed by the system for coordinators from the knowledge base. If the user chooses one of proposed activities, corresponding situation of actor who will be in charge of this activity will be generated, such as type of actor, number of actor, duration of activity. And if the coordinator prefers to create the new activity, they can input all the information of activity manually. The main operation of activity table is shown in Table 3-9.

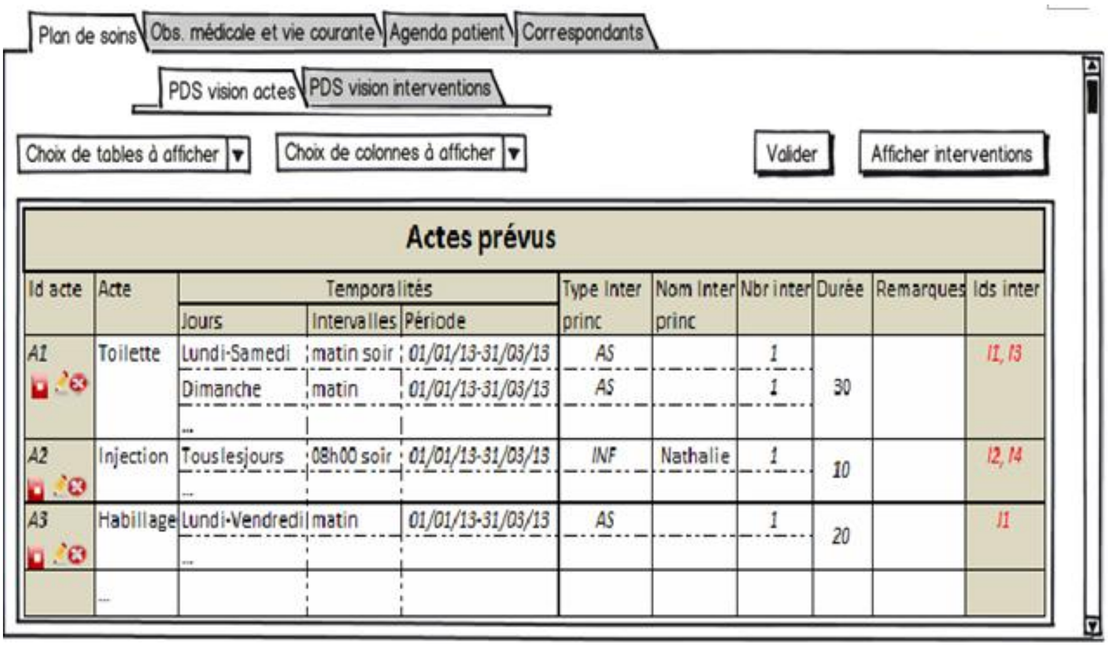
The most important function of this table is to specify the temporality for each activity. After inputting activity name for each patient, the medical coordinator will specify the temporality using one new window (see Figure 3-14). For one temporality, it will be selected in the sequence "period->days->time ranges". The default period will be proposed by system and it could be modified by user as he wants. For days, in total, there are seven conditions: Every day (all the day during the whole period), detailed date (any date during the whole period), some days (one or more day during the week), even days, odd days, holidays (especially French holidays) and every N days (such as every 10 days). For time ranges, there are 3 conditions: Nothing (any time), intervals (Moring, noon, afternoon, evening, night) and detailed time (such as 08H00 AM).

Table 3-9 The main operation of activity table

Operation	**Description**
Choose the column of table	The user could select which column they need column by selecting list

Table 3-9 The main operation of activity table (continued)

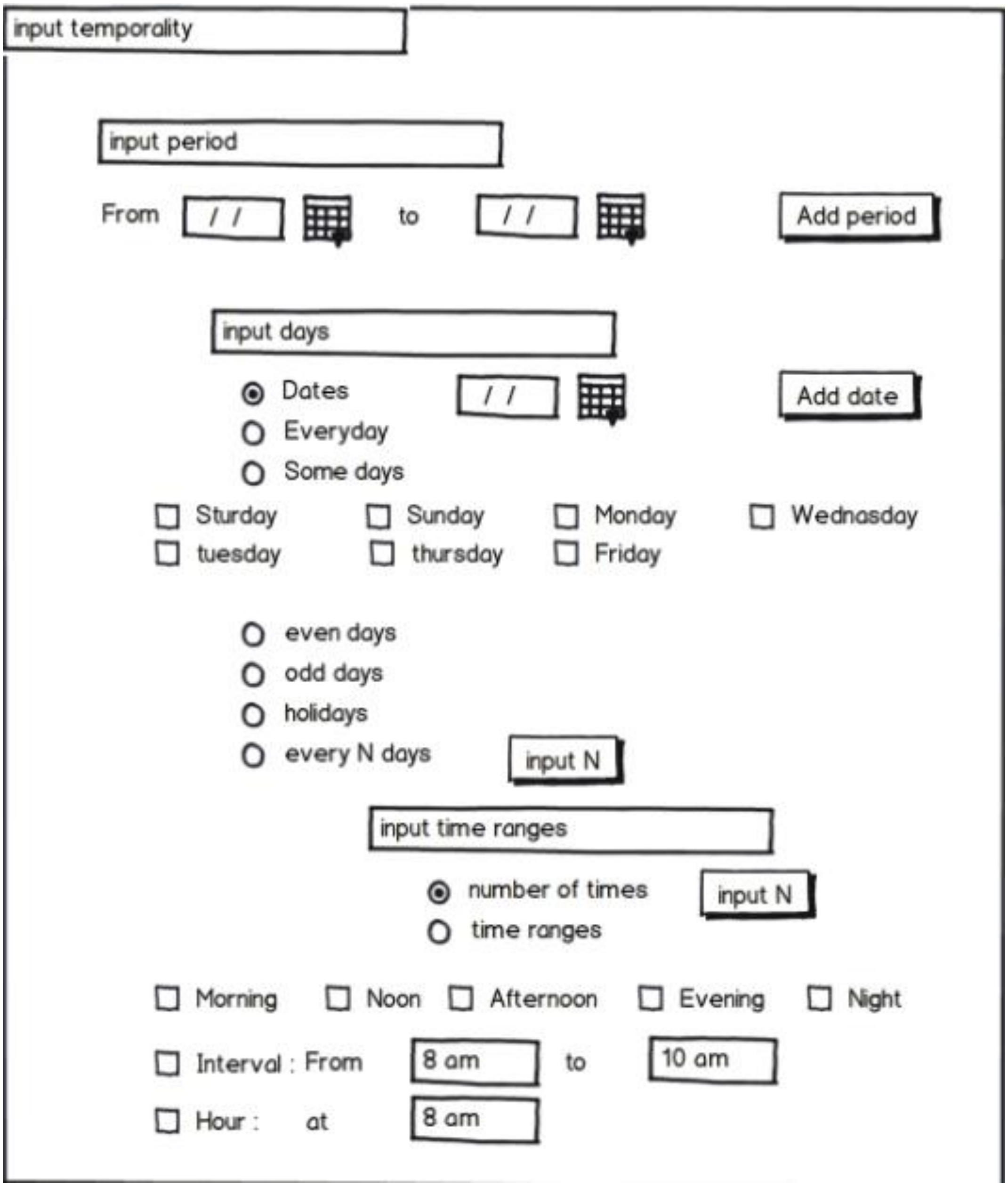
Operation	Description
Modify the content of table	The user could modify the content of table manually. System will give the error prompt if the user input illegal information
Commit the content of table	After the operation of users, all the information could be stored into database by clicking button ‘Valider’
Delete one of the activities	If medical coordinator wants to delete one of activity, he could click the ‘delete’ icon to delete one row of activity table. But the ID of activity will not be changed, for example, A2 will not become A1 automatically after deleting A1.
Specify the temporality for each activity	Each activity will have one temporality
Generate the intervention table	By clicking the button ’Afficher interventions’, the activity table will be transformed to the intervention table
Display/hidden the agenda table	By clicking the button ’Afficher interventions’, the agenda table will be switched between “display” and “hidden” state



Actes prévus

Id acte	Acte	Temporalités			Type Inter	Nom Inter	Nbr inter	Durée	Remarques	Ids inter
		Jours	Intervalles	Période	princ	princ				
A1	Toilette	Lundi-Samedi	matin soir	01/01/13-31/03/13	AS		1	30		I1, I3
		Dimanche	matin	01/01/13-31/03/13	AS		1			
		...								
A2	Injection	Touslesjours	08h00 soir	01/01/13-31/03/13	INF	Nathalie	1	10		I2, I4
		...								
A3	Habillage	Lundi-Vendredi	matin	01/01/13-31/03/13	AS		1	20		I1
		...								
	...									

Figure 3-13 Activity table in the making home care plan module



input temporality

input period

From / / to / / Add period

input days

● Dates / / Add date

○ Everyday

○ Some days

□ Sturday □ Sunday □ Monday □ Wednasday

□ tuesday □ thursday □ Friday

○ even days

○ odd days

○ holidays

○ every N days input N

input time ranges

● number of times input N

○ time ranges

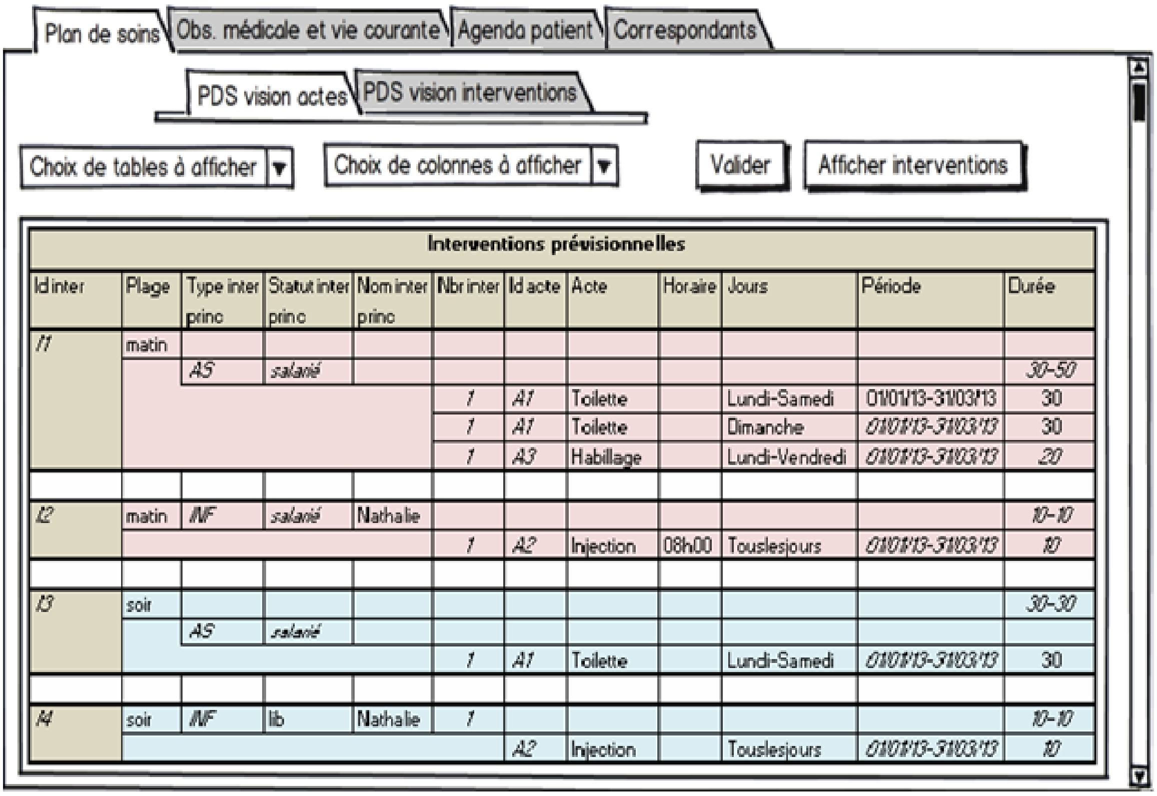
□ Morning □ Noon □ Afternoon □ Evening □ Night

□ Interval : From 8 am to 10 am

□ Hour : at 8 am

Figure 3-14 the temporality selected window for each activity

- In addition to the notion of activity, another important concept of the DSL lies in the notion of Intervention. An intervention is a collection of activities that can be scheduled together. Interventions are defined by grouping together the activities that can be performed by a same professional and which occur in the same time range. Interventions may be specified manually by the coordinator or computed automatically from the specifications of the activities and then proposed to the coordinator for validation. Figure 3-15 shows the GUI corresponding to the interventions. Besides the aforementioned concepts, the proposed DSL is enriched with additional constraints derived from the domain knowledge (e.g., medical knowledge represented in ontologies, etc.). For example, various types of dependencies between activities may exist such as: 1) Obligation to perform a given activity in a time period after a first one. For example, "a Lovenox injection must be followed by a blood test within a time limit of one week".2) For the exclusion of an activity in a given period, it will be restricted. For example, "a minimum of 12 hours is requested between two insulin injections".



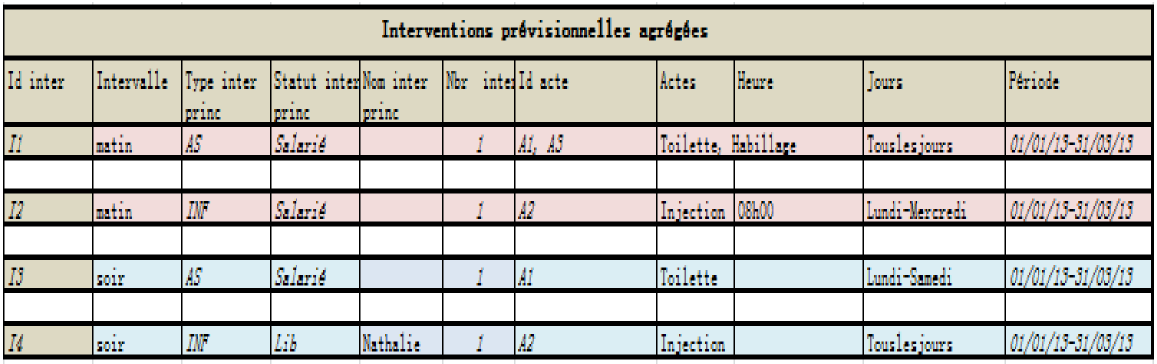
Interventions prévisionnelles

Id inter	Plage	Type inter princ	Statut inter princ	Nom inter princ	Nbr inter	Id acte	Acte	Horaire	Jours	Période	Durée
I1	matin										
		AS	salarié								30-50
					1	A1	Toilette		Lundi-Samedi	01/01/13-31/03/13	30
					1	A1	Toilette		Dimanche	01/01/13-31/03/13	30
					1	A3	Habillage		Lundi-Vendredi	01/01/13-31/03/13	20
I2	matin	INF	salarié	Nathalie							10-10
					1	A2	Injection	08h00	Touslesjours	01/01/13-31/03/13	10
I3	soir										30-30
		AS	salarié								
					1	A1	Toilette		Lundi-Samedi	01/01/13-31/03/13	30
I4	soir	INF	lib	Nathalie	1						10-10
						A2	Injection		Touslesjours	01/01/13-31/03/13	10

Figure 3-15 Intervention table in the classifying home care plan module

This table is one transformation of activity table, with similar function and content. It consists of ID of intervention, time ranges, type of actors, number of actors, name of actors, corresponding ID of activity, name of activity, hour, days, periods and duration. The home care plan is classified by time range and type of actors together. The duration of intervention is the range calculated by the duration of corresponding activities. For instant, in the given example, the intervention I1 includes A1 (duration: 30 days) and A3 (duration: 20 days). So the duration of I1 is the period from the maximum (A1, A3) to the summation of A1 and A3.

Other format of intervention table is shown in figure 2-8. It consists of ID of intervention, time ranges, type of actors, number of actors, name of actors, corresponding ID of activity, name of activity, hour, days, periods and duration. It presented the main information instead of detailed information. The main purpose of this table is to create the connection between id of intervention and id of activity. It will simplify the transformation between intervention table and activity table.



Interventions prévisionnelles agrégées

Id inter	Intervalle	Type inter princ	Statut inter princ	Nom inter princ	Nbr inte	Id acte	Actes	Heure	Jours	Période
I1	matin	AS	Salarié		1	A1, A3	Toilette, Habillage		Touslesjours	01/01/13-31/03/13
I2	matin	INF	Salarié		1	A2	Injection	08h00	Lundi-Mercredi	01/01/13-31/03/13
I3	soir	AS	Salarié		1	A1	Toilette		Lundi-Samedi	01/01/13-31/03/13
I4	soir	INF	Lib	Nathalie	1	A2	Injection		Touslesjours	01/01/13-31/03/13

Figure 3-16 Another intervention table in the classifying home care plan module

3.4.3 Design of temporality selected interface

According to requirements, the home care plan presents an activity table in the user layer. Meanwhile, temporality selected module is the key module of an activity table, which focuses on designing and implementing the feasibility of specifying home care plan. Temporality consists of three parts: period, days, time ranges. The default period is proposed by the knowledge base. The format of period is that from Day/Month/Year to Day/Month/Year. For the attribute ‘days’, there are four options to provide for the end user: detailed date, every day, some days and festivals. The end user will choose one of them as his option. Detailed date is one or more dates selected during the period. Meanwhile, every day means that specifying each day during this period. For some days, the end user will choose one or more days during the whole week. What is more, if the end user chose continuous 3 or more days, the result of days will be displayed in the form of “xx-xx”. For example, if the user choose Monday, Thursday, Friday and Saturday using multiple checkbox. The result will be shown like “Monday, Thursday - Saturday”. The algorithm is designed as following:

```
String[] str = the set of selected days
// switch days during weeks to the number {1-7};    {1,3,4,5,7}
Switch(str,strnumber);
sort ascending-> strnumber;
List List_single,List_start,;
Flag = 0;
For(i = 0; i<strnumber.length;i++)
{
    If(strnumber[i]+1 = =strnumber[i+1]){
       Flag++;
    }
    else{
    flag =0;
    }
    If(flag>1)
    List_start<- flag;
    List_end<- i; }
    Else{
          Flag=i+1;
        List_single<-i;}
}
```

For the festival in the France, our time concludes the whole festivals in the Table 3-10, because all of them will be fix date, they will be stored into rational database.

For time ranges, there are two conditions: nothing and intervals. Nothing stands for any time in the one day and intervals contain six situations: morning (7h00-11h00), noon (11h00-13h00), afternoon (13h00-18h00), evening (18h00-22h00), night (22h00-7h00 at tomorrow) and detailed time such as 06h57. The end user could select one or more options on intervals. If detailed time has conflict with other intervals, for example, detailed time is 10h17 and interval is at morning, the system will judge detailed time whether is in the intervals.

Table 3-10 Festivals in France

Festival Name	Date
New Year's Day	1 January
May Day	1 May
Victory in Europe Day	8 May
Bastille Day	14 July
Assumption of Mary to Heaven	15 August
All Saints' Day	1 November
Veterans Day	11 November
Christmas Day	25 December
St. Stephen's Day	26 December

After the temporality was selected, the user could preview the temporality table which is the same information in the activity table. The temporality table is able to have drug and drop function. We use BeanItemContainer to save tableContainer which corresponds to "Temporality.class". Each time user could choose drug the one line (a temporality situation) to order the temporality for the coordinator.

```
BeanItemContainer<Temporality> tableContainer = new
BeanItemContainer<Temporality>(Temporality.class);
dpp.getTable_temporality().setContainerDataSource(tableContainer);
dpp.getTable_temporality().setDragMode(TableDragMode.MULTIROW);
```

When the user drags one line to the top or middle of another line, it means that he would like to move it above another line. Otherwise, it means that move it blow another line. Meanwhile, user could delete one of the temporality. The whole process is shown in the detail.

```
switch (location) {
        case BOTTOM:
                tableContainer.removeItem(sourceItemId);
                tableContainer.addItemAfter(targetItemId, sourceItemId);
                break;
        case MIDDLE:
        case TOP:
                tableContainer.removeItem(sourceItemId);
                final Object prevItemId = tableContainer.prevItemId(targetItemId);
                tableContainer.addItemAfter(prevItemId, sourceItemId);
                break;
}
```

3.4.4 Design of intervention table interface

Intervention table is a reclassification of an activity table according to type of actors and time restriction. Intervention table module is divided into two sub-modules: Intervention developed module and Intervention simplified module. Developed table is described in detail that all the information of the activity table will be shown and simplified table is defined that the important information will be shown. The purpose of intervention table is that user could verify the reasonableness of home care plans because it will be clearer that activity table. User could choose two kinds of table to view the situation of classification through the drop list. Intervention developed module corresponds to InterventionDevelopee class which cooperates with other sub-modules and implements classification of activity table. The data in this module provides an easy-to-understand model for the users. According to different time ranges and type of actors, the intervention developed table will show the information in different order.

In the database, all the data on time restricts will be transformed to detailed time in unit of hours. For example, “Monday-Thursday” will be transformed into $4\times24=96$ hours. Firstly, the home care plan is classified by days, such as morning or afternoon. Then the result will be classified by the type of actors again. The purpose of the process is to find which activities correspond to the different intervention. An intervention includes several activities, so we need calculate the duration of each intervention according to the duration of activities. We assume that one intervention I_1 consists of N activities ($A_1 \ldots A_n$). The duration of A_i is D_i days. As the result, the range of duration of I_1 is $(\max\{D_1,\ldots,D_n\},\sum_1^n D_i)$. We define that the Classifier class is in charge of allocating the home care plan. The output information includes 1) ID of intervention, 2) days, 3) type of intervention, 4) status of actor, 5) numbers of actor, 6) name of actor, 7) temporality of each intervention and 8) duration in total.

For the Intervention simplified module, it is a simplified intervention table which only cares that which activities maps the intervention.

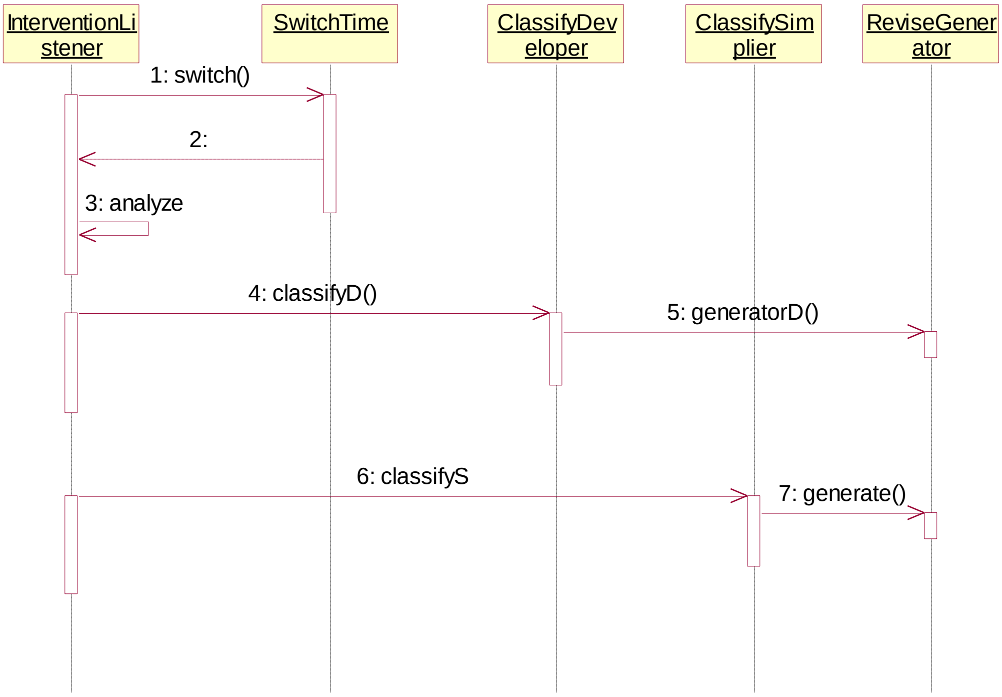
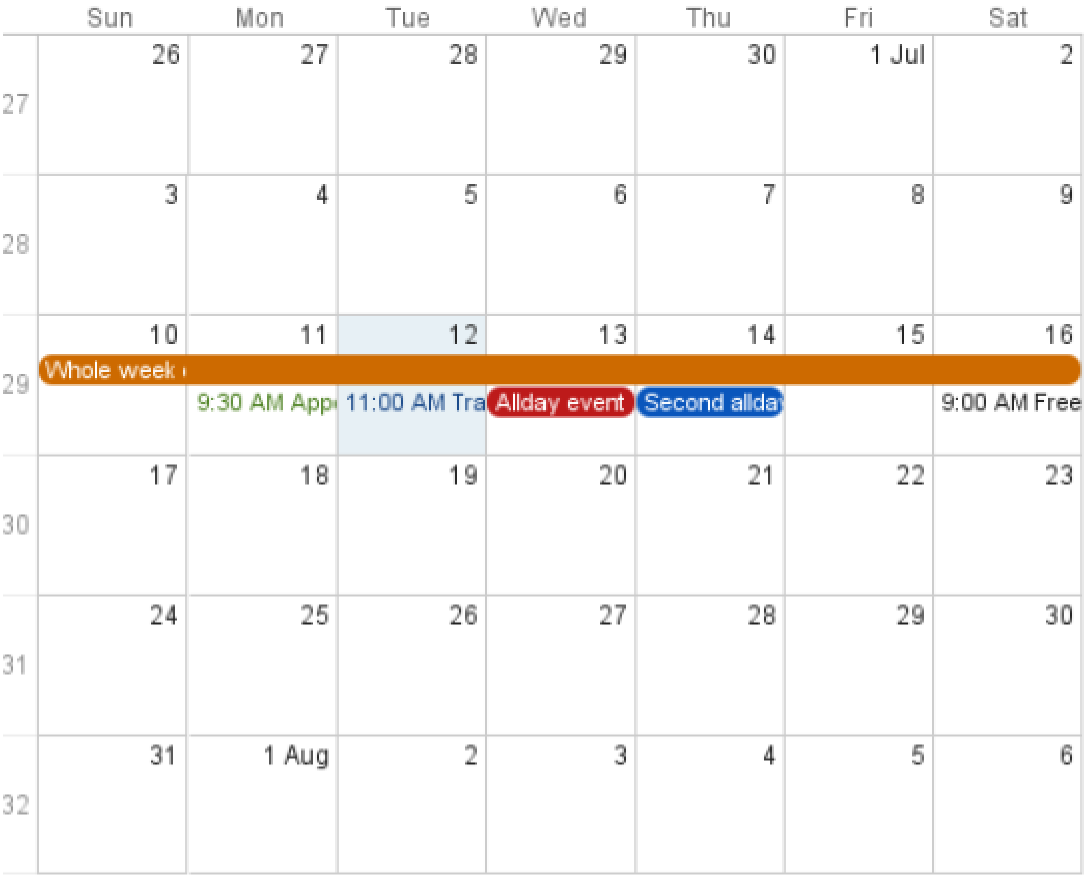


Figure 0-16 Classifying intervention table sequence diagram

Figure 0-26 illustrates the sequence diagram in classifying intervention table. InterventionListerner call switch() method to translate data into hours. After getting the result, analyze() method will compare different time ranges each other to find the activities that has same time ranges to save into various vectors. And then by classify() method, these activities will be classified again according to the same type of actors.

3.4.5 The agenda of home care plan

In the main page, there is a button to display or hidden the agenda view of home care plan. We use Vaadin Calendar to implement it. For the temporality of the activity, we could transform the triple {period, days, time ranges} to detailed date and detailed hours, for instance, {01/01/2013-15/01/2013, Sunday, morning} is transformed to 04/01/2013 and 11/04/2013. Then by the Calendar class to configure the detailed date, different color will be used to express different meaning.



	Sun	Mon	Tue	Wed	Thu	Fri	Sat
27	26	27	28	29	30	1 Jul	2
28	3	4	5	6	7	8	9
29	10 Whole week	11 9:30 AM App	12 11:00 AM Tra	13 Allday event	14 Second allda	15	16 9:00 AM Free
30	17	18	19	20	21	22	23
31	24	25	26	27	28	29	30
32	31	1 Aug	2	3	4	5	6

Figure 0-17 Monthly views with All-Day and Normal Events

Figure 3-17 shows that home care plan will be remarked according to the date of home care plan. Different activities have various colors. Following are the different colors of activity name. 1) Green color, the activity should be done but the actor will not have free time to finish it. 2) Blue color. The activity should be done but the actor will be able to finish it .3) Red color. The activity could be done at any time during the whole day. 4) Yellow color. The activity could be done at any time but there is another conflict with other activities. 4) No color. The activity could not be done at any time and the coordinator need modify it.

3.5 Design of logic layer

In this section we present our approach to map care plans into timed automata. We propose a three steps approach which works as follows: (i) firstly, for home care plans, we need analyze and monitor them; (ii) secondly, basic activities of a care plan are translated into timed automata using an appropriate algorithm; then (iii) using UPPAAL to check model.

Analysis and monitoring of care plans. Once care plans are constructed using the DSL, they can be exploited for two purposes: static analysis of care plans and monitoring of care plans executions.

Such an analysis is performed at the design time, i.e., without actually executing the care plans, and targets the verification of various types of properties, such as:

1) Care plans verification. This analysis takes place within a specific care plan and aims at checking all the possible run-time errors that may occur in the considered plan. For example, it is important to check the reliability of a care plan, i.e., to check whether or not the activities included in the plan can be effectively scheduled and performed according to the constraints specified in the plan.

2) Interventions verification. This analysis takes into account interactions between the temporal constraints of an intervention and the constraints of the activities included in the considered intervention. For example, if an activity A appears in intervention I, then it is worth to check whether the temporal constraints of A are compatible with those of I.

3) Consistency verification. In some cases a specification must satisfy some dependencies between activities. For example, "a Lovenox injection must be followed by a blood test within a time limit of one week" or "a minimum of 12 hours is requested between two insulin injections". Indeed, a dependency between two activities entails dependencies between the corresponding Interventions. Hence, the verification of consistency must be extended to interventions.

4) Compatibility verification. This analysis takes into account the patient agenda in order to avoid making plan for activities in time slots in which the patient is not available.

3.5.1 Formal framework based on timed automata

This section first recalls some basic notion from the theory of timed automata then it illustrates the mapping of care plans into timed automata.

Definition 1.(Timed automata) A timed automata[22] is a tuple $A=(S,s_0,\sum,X,Inv,T,F)$ where:

1) S is a finite set of locations or states of the automaton and $F\subseteq S$ is a set of final states;
2) $s_0\subseteq S$ is a set of initial locations;
3) $\sum$ is a finite set of transition labels;
4) X is a finite set of clocks;
5) $Inv:S\rightarrow\phi(X)$ associates an invariant to each state of the automaton;
6) $T\subseteq S\times\sum\cup\{\varepsilon\}\times\phi(X)\times 2^X\times S$ is a set of transitions. A transition (s,a,ϕ,λ,s') represents an edge from location s to location s' on symbol a. ϕ is a clock constraint, and the set $\lambda\subseteq X$ gives the clocks to the reset after firing such a transition.

Each activity a set of a care plan is mapped into a timed automaton A_a. The construction of A_a depends on the specification of the activity a and in particular its associated time constraints. We explain the mapping of a basic activity into a timed automaton using a simple example. Consider the following specification of a basic activity: {Activity name, days, time ranges, period}.

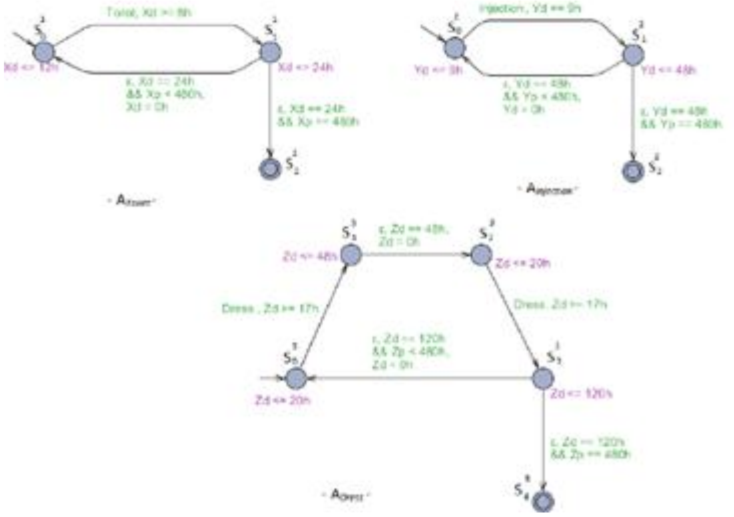

Toilet, Xd >= 8h
Xd <= 12h
Xd <= 24h
ε, Xd == 24h && Xp < 480h, Xd = 0h
ε, Xd == 24h && Xp == 480h
Injection, Yd == 9h
Yd <= 9h
Yd <= 48h
ε, Yd == 48h && Yp < 480h, Yd = 0h
ε, Yd == 48h && Yp == 480h
ε, Zd == 48h, Zd = 0h
Zd <= 48h
Zd <= 20h
Dress, Zd >= 17h
Dress, Zd >= 17h
ε, Zd == 120h && Zp < 480h, Zd = 0h
Zd <= 20h
Zd <= 120h
ε, Zd == 120h && Zp == 480h

Figure.3-18. The result of the mapping of the basic activities of the care plan

The corresponding timed automata $A_{activity} = (S, s_0, \Sigma, X, Inv, T, F)$, which is depicted at figure 3-18, is defined as follows:

$S = \{s_0^1, s_1^1, s_2^1\}$ is the set of states with s_0^1 the initial state of $A_{activity}$;

$F = \{s_2^1\}$ is the set of final states;

$\Sigma = \{Activity\} \cup \{\varepsilon\}$ is the set of transition labels;

$X = \{x_d, x_p\}$ is the set of variables where the variables where the variable x_d is used to measure the flow of time during a day and x_p is the variable used to control the whole period. The variables x_d and x_p are expressed using as time unit the hour,

T is the set of the following transitions:

1) $(s_0^1, Activity, x_d \geq 8, \phi, s_1^1)$, this transition specifies that when the automata are at state s_0 , it can moves to state s_1 upon the execution of the activity. The conjunction of the state invariant $x_d \leq 12$ at s_0^1 and the transition guard $x_d \geq 8$ ensure that the activity can be performed only in the morning (i.e., when the value of x_d is between 8 and 12).

2) $(s_1^1, \varepsilon, x_d == 24 \wedge x_p \leq 480, \{x_d\}, s_0^1)$, this transition enables the automata to move back from s_1^1 to s_0^1, without performing any activity, at the end of the day (i.e. when x_d equals to 24) with the specified period (i.e. $x_p \leq 480$). Upon this transition, the variable x_d is reset to 0 to record the beginning of a new day.

3) $(s_1^1, \varepsilon, x_p == 480, \phi, s_2^1)$, this transition is fired at the end of the period (i.e. when x_p equals to 480) and it enables the automata to move to the final state s_2^1 and terminate the execution.

In addition to the automaton Activity, figure 3-18 the following two additional automata constructed in a similar way: The timed automaton $A_{Injection}$ corresponding to the activity "Injection every two days at 9 am from 06/10/13 to 06/30/13", and the timed automaton A_{Dress} corresponding to the activity "Dress on Monday and Wednesday evening from 06/10/13 to 06/30/13".

Interestingly, the care plan can be also described by means of timed automata which are obtained by composition of automata representing basic activities that have been generated in the first step. The composition is achieved using the asynchronous product of activities automata which allow us to recognize all possible configurations of the care plan. We recall below the definition of asynchronous product (or shuffle) of timed automata.

Definition 2. (Shuffle of timed automata) Let $A_1 = (S_1, s_0^1, \sum_1, X_1, Inv_1, T_1)$ and $A_2 = (S_2, s_0^2, \sum_2, X_2, Inv_2, T_2)$ be two timed automata. The product of A_1 and A_2, donated $A_1 \times A_2$, is the timed automata $(S_1 \times S_2, s_0^1 \times s_0^2, \sum_1 \cup \sum_2, X_1 \cup X_2, Inv, T)$, where $Inv(S_1, S_2) = Inv(S_1) \wedge Inv(S_2)$ and the transition function T is defined as follows: T={ $((s_1, s_2), a, \phi, \lambda, (s'_1, s'_2)) : ((s_1, a, \phi_1, \lambda_1, s'_1) \in$ T_1 and $s_2 = s'_2$ or $(s_2, a, \phi_2, \lambda_2, s'_2) \in T_2$ and $s_1 = s'_1)\}$.

The care plan is composed of three activities, each with temporal modalities: (1) "Toilet every day in the morning"; (2) "Injection every day at 9h00" and (3) "Dress every day in the morning". Figure 3-18 shows the automaton representation for each activity, where A_{Toilet}, $A_{Injection}$and A_{Dress} represent the timed automata corresponding to the three activities: toilet, injection and Dress.

Figure 3-19 shows the product of the automata A_{Toilet} and $A_{\text{Injection}}$ of figure 5. The resulting automaton encompasses all the possible schedules of the activities Toilet and Injection.

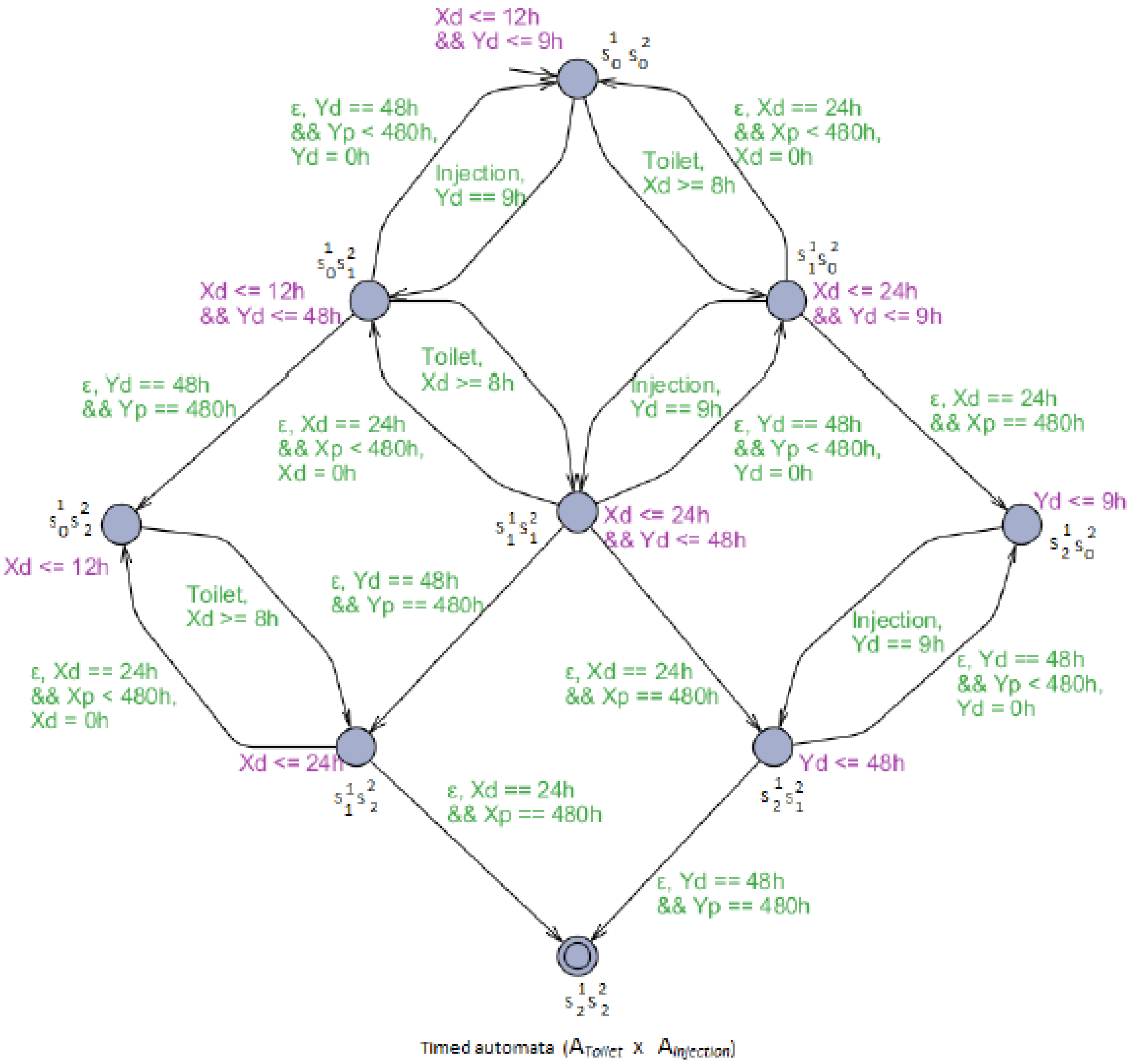


Figure 3-19 Intervention of every day morning from 06/10/13 to 06/30/13

3.5.2 Formal analysis of care plans using timed automata

With a formal model describing the behavior of care plans at hand, it becomes now possible to handle automatic verification and monitoring of care plans. We discuss below how the proposed framework can be used to automate generation of Interventions and verification and monitoring of care plans.

- Automatic generation of candidate interventions automata. Given a time specification T, we may want to generate the interventions corresponding to T. The specification T may be defined as a precise time range (e.g. Intervention of the morning, intervention of the evening, etc.); a combination of days and time ranges (e.g. Intervention of the Monday morning, Wednesday afternoon, etc.) or a combination of time ranges, days and period (e.g. Interventions of every Monday morning from 03/10/13 to 30/10/13, etc.). Candidate interventions are generated automatically using a composition of activity automata projected on the time specification T. For example (see figure 3-18) depicts an automaton representing an intervention that is carried out every day, morning, from 06/10/2013 to 06/30/2013. This automaton is obtained by first projecting the activity automata A Toilet and An Injection of figure 3-18 with respect the appropriate time specification (i.e., Time range = morning, Days = every day and Period =[06/10/2013-06/30/2013]) and then computing the asynchronous product of the result.

- Care plan verification. For example, checking reliability of a care plan can be reduced to the emptiness problem of the corresponding automaton.
- Intervention verification. Interventions are formally described as automata (in the same spirit of the previous mapping). Then intervention verification is achieved by checking the emptiness of the intersection of the activity automata with the intervention automata.
- Consistency verification. Dependencies are expressed as automata and then consistency checking is formulated as a composition of interventions automata with dependency automata.
- Monitoring. Checking compliance of executions traces can be reduced to language recognition in timed automata. Moreover, it is possible to extend the proposed automata to manage alarms. For example, a care plan automata can be used to enforce that an alarm has to be activated one week after a Lovenox Injection has been performed.

3.6 Key techniques

The Key techniques of the new approach include transformation from meta-model to rational database, user interface layer design based on DSL and implement of model verification.

Transformation from meta-model to rational database combines traditional code generation technique with object-relationship mapping technique and modify some process of them to meet requirements of research.

The user interface layer is designed and implemented by the Vaadin framework. It aims to improve the efficiency of the primary medical coordinator and simplify the whole process of specifying home care plan. Although the business process modeling could to provide the home care plan specifying home care plan, home care plans can be viewed as a kind of non-structured business processes which, as a consequence, cannot be managed using standard BPM technology. So we focus on the problem of designing and implementing a Domain Specific Language to facilitate modeling, execution and monitoring home care plans. This is a timely and challenging problem. In the home care plan specifying process, the main two contents are patients' information and home care plan of this patient which consists of activity table and intervention table. In the DSL method, formulating the temporality is the core of the whole process. The temporality consists of a triple (period, days, time ranges) which is the input of automata generation.

Model verification technique in our approach aims to monitor and verify the home care plan making process. We define the timed automata model and use the temporality triple as the input to generate timed automata. The model checker UPPAAL is used to detect and modify the automata in order to monitor and give the prompts to specify home care plan. It is very useful for the end user to manage and specify home care plan and improve the feasibility of home care plan.

3.7 Brief summary

The chapter has introduced the whole layered architecture of the novel approach we put up with and the detailed design of each layer. The architecture of the project is divided into three layers: database layer, graphical user interface layer and logic layer. The database layer has described how to transformation from the meta-model in form of class diagrams to the rational database. It is mainly divided into two steps: firstly transformation between class diagrams and java code; connection between java code and rational database. The interface layer has defined the whole process to specify home care plan and detailed design of the intelligent tool to simplify and visualize the home care plan specifying process. It consists of the design of patient information panel, the design of activity table, the design of intervention table and how to switch between activity table and intervention table. Finally, the logic layer collects the information on the temporality of each home care plan and design the timed automata model, and then we use the UPPAAL to generate automata and verify it. The final purpose is to modify the home care plan and give the prompts to the end user.

Chapter 4 System Implementation and Testing

This chapter will focuses on the implementation of this project, including the realization of automatically generation of database and the implementation of graphical user interface, as well as the testing of software. For the implementation of software needs to utilize software engineering, this chapter will firstly introduce project's development and implementation environment and then state how to implement the three layers in the project. Finally the project will be well tested.

4.1 The environment of system implementation

This project is developed based on a PC with Microsoft 7 SP2 as Operating System. The computer configurations listed below:

CPU: Mobile Dual Core Intel Core i7-2637M, 2500 MHz (25 x 100),

System memory: 3964 MB (DDR3 SDRAM),

Motherboard: Intel Direct Media Interface v2.0

Following is a list of software development environment of project:

- Programming Language: Java, domain-specific language;
- Development environment: Eclipse Juno;
- Database: MySQL;
- Rich internet application framework: Vaadin 7;
- ORM Tool: Hibernate framework;
- Modeling Verification Environment: UPPAAL;
- Code Generation Tool: Rational Rose IBM, MagicDraw.

4.2 Implementation of Database layer

4.2.1 Transformation of meta-model to java codes

Firstly the meta-model will be established by Rational Rose in form of class diagram. For each class, we will add one attribute 'id' to class as its unique identity. After that, we will set the configuration as follows:

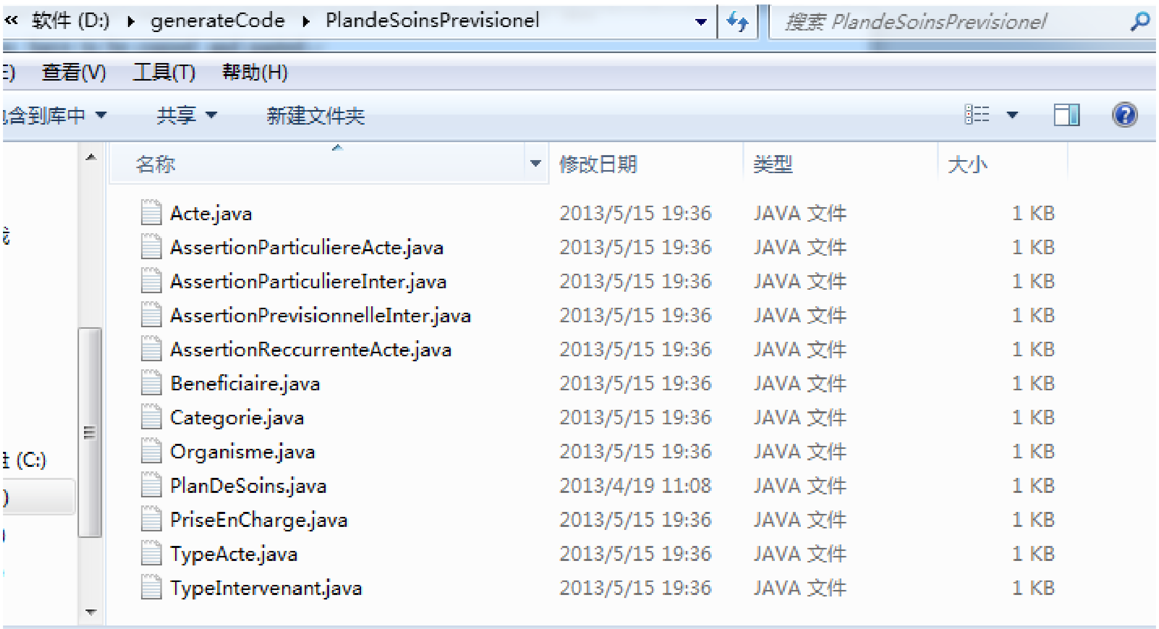
Firstly, in the Rational Rose, open the designed class diagrams and choose the packages, component or class which we need transform. Rational Rose code generation generate once all the classes which are in the logical view or component view. In the Class Path option, all the related directories should be in the Environment Settings box. For our project, all the objects should be added into the user interface package.

Secondly, Rational Rose has model verification function independently on the various types of programming languages. And this function is used for checking the syntax of the class diagrams to assure the consistency of model. It is better that check the syntax of the class diagrams before generating codes. The operation on the model checker in the Rational Rose is to select” Tools” option and then click the “Check Model” to verify the correctness of the model. Meanwhile, you could take the same operation by selecting “Tools” option and then click the “Syntax Check” to verify the correctness of syntax of the model.

Thirdly, we can configure the code style such as Javadoc style, author name and so on. By selecting “Tools” option and then click the “Options” to define the attributes generated by class.

At last, user could select “Tools” option and click “Class classification” and then select the option ”Generate Java Code”. If the java package exists, the generated code will cover the original one and not delete the java class that has been deleted in the class diagrams.

Figure 4-1 shows the result of the transformation process from meta-model to java codes. Each class will generate corresponding java class. In each java class, for example, in the class Acte (see in the figure 4-2), the source file is the directory of this file. Each attribute has its Get and Set functions.

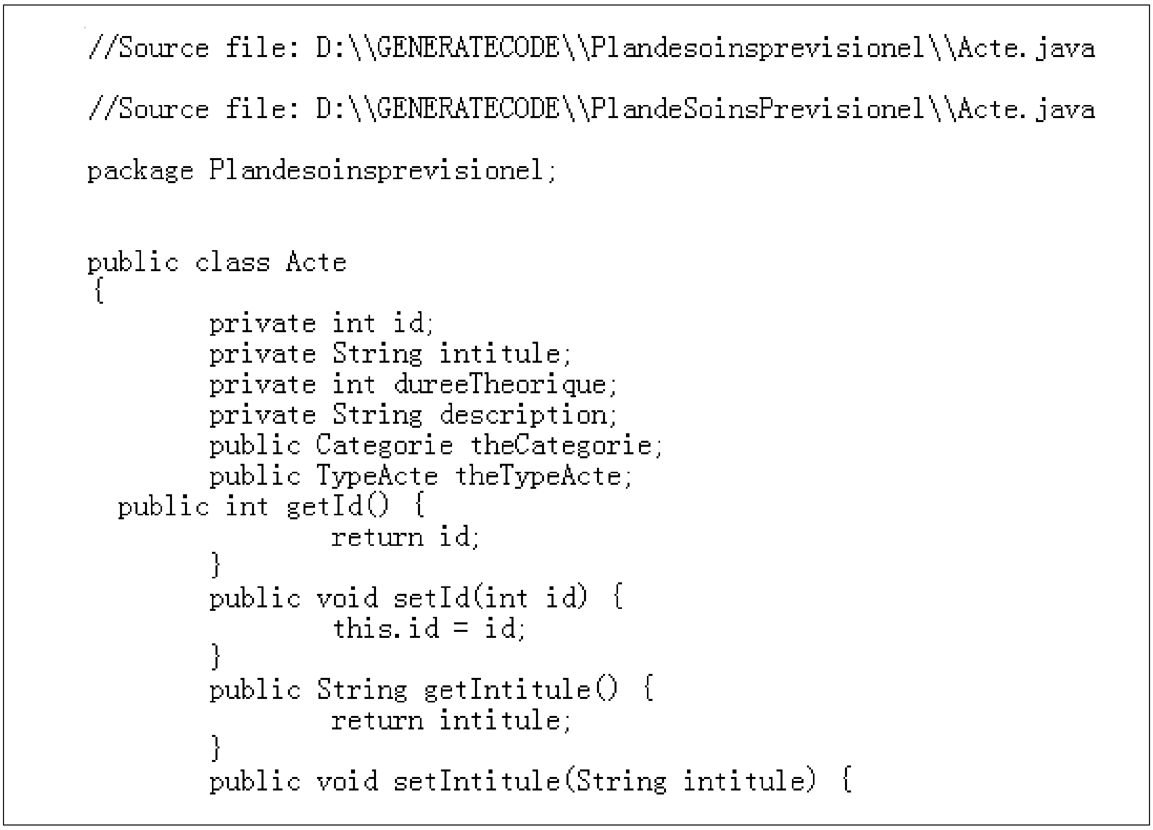

« 软件 (D:) ▸ generateCode ▸ PlandeSoinsPrevisionel
搜索 PlandeSoinsPrevisionel
查看(V)
工具(T)
帮助(H)
含到库中
共享
新建文件夹
(C:)

名称	修改日期	类型	大小
Acte.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionParticuliereActe.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionParticuliereInter.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionPrevisionnelleInter.java	2013/5/15 19:36	JAVA 文件	1 KB
AssertionReccurrenteActe.java	2013/5/15 19:36	JAVA 文件	1 KB
Beneficiaire.java	2013/5/15 19:36	JAVA 文件	1 KB
Categorie.java	2013/5/15 19:36	JAVA 文件	1 KB
Organisme.java	2013/5/15 19:36	JAVA 文件	1 KB
PlanDeSoins.java	2013/4/19 11:08	JAVA 文件	1 KB
PriseEnCharge.java	2013/5/15 19:36	JAVA 文件	1 KB
TypeActe.java	2013/5/15 19:36	JAVA 文件	1 KB
TypeIntervenant.java	2013/5/15 19:36	JAVA 文件	1 KB

Figure 4-1 the output of the code generation

4.2.2 Transformation of java code to relational database

After java class is generated automatically, we use ORM (Object-Relational Mapping)/Hibernate framework to create the connection between java code and relational database. User can use Hibernate which generates the SQL on the fly and then automatically executes the necessary SQL statements. This saves a lot of development and debugging time of the developer. Writing JDBC statement, setting the parameters, executing query and processing the result by hand is lots of work. Hibernate framework will save all tedious efforts. Hibernate is layers architecture and you can use the components as per your application need. The whole process is described as figure 4-3.



```
//Source file: D:\\GENERATECODE\\Plandesoinsprevisionel\\Acte.java

//Source file: D:\\GENERATECODE\\PlandeSoinsPrevisionel\\Acte.java

package Plandesoinsprevisionel;


public class Acte
{
        private int id;
        private String intitule;
        private int dureeTheorique;
        private String description;
        public Categorie theCategorie;
        public TypeActe theTypeActe;
  public int getId() {
                return id;
        }
        public void setId(int id) {
                this.id = id;
        }
        public String getIntitule() {
                return intitule;
        }
        public void setIntitule(String intitule) {
```

Figure 4-2 java codes of the Acte class

Mapping Java classes to database table is accomplished through the configuration of an XML file or by using Java Annotation. When using an XML file, Hibernate can generate skeletal source code for the persistence classes. This is unnecessary when annotation is used. Hibernate can use the XML file or the annotation to maintain the database schema. Facilities to arrange one-to-many and many-to-many relationships between classes are provided. In addition to managing association between objects, Hibernate can also manage reflexive associations where an object has a one-to-many relationship with other instances of its own type. Hibernate supports the mapping of custom value types. This makes the following scenarios possible: 1) Overriding the default SQL type that Hibernate chooses when mapping a column to a property; 2) Mapping Java Enum to columns as if they were regular properties; 3) Mapping a single property to multiple columns.

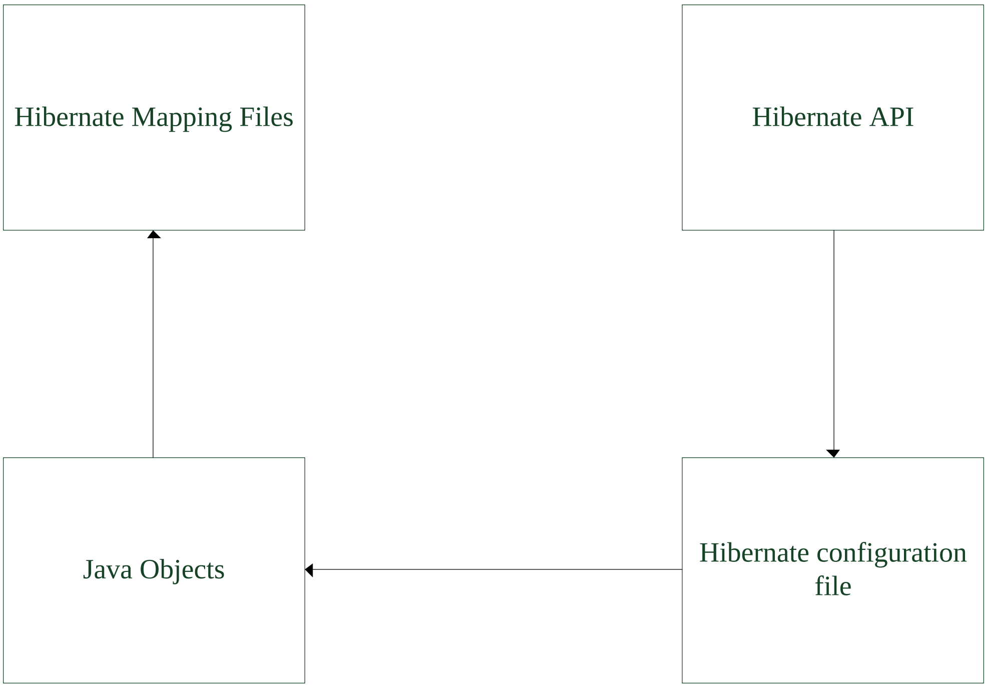


Figure 4-3 The whole process of using hibernate

1.1.1.4 4.2.2.1 Hibernate configuration

Firstly, the normal java project will be created. Because a new hibernate library is useful and convenient for hibernate project. The user could go to Eclipse IDE to choose "windows -> preference->java->build path->use libraries" in order to create a new libraries "hibernate3". The three libraries will be added into "hibernate3": the folder 'lib', hibernate3.jar and jdbc(mysql driver). After the new library was created, you should refer to it as following: firstly select the 'preference' and then go to 'java build path'. Though choosing 'add library' to click 'user library'.

1.1.1.5 4.2.2.2 Hibernate mapping files

There are five situations between two classes in the java code using hibernate: one-to-one association (one-to-many is a special association of one-to-one association), many-to-many association, aggregation, inheritance (sub classes) and recursive association. By convention, Hibernate XML mapping files are named with the .hbm.xml extension.

For the one-to-many and many-to-many association, we write the hbm.xml to implement the relationship as following:

```
     <many-to-one name="thePriseEncharge" column="priseenchargeid"/>
      <many-to-one name="theBeneficiaire" column="beneficiaireid"/>
      <set name="theActePrevus">
      <key column="plandesoinsid"/>
      <one-to-many class="PlandeSoinsPrevisionel.ActePrevu"/>
     </set>
     <set name="theInterventionPrevisionnelles">
               <key column="plandesoinsid"/>
          <one-to-many class="PlandeSoinsPrevisionel.InterventionPrevisionnelle"/>
               </set>
```

For the inheritance mapping, we will pick up all the attributes of the parent class to add into the sub-class. The hbm.xml is described as following:

```
<hibernate-mapping package="domain">
     <class name="Employee" discriminator-value="0">
          <id name="id">
               <generator class="native"/>
          </id>
          <discriminator column="type" type="int"/>
          <property name="name"></property>
          <many-to-one name="dpt" column="dpt_id" lazy="false"></many-to-one>
          <subclass name="Skiller" discriminator-value="1">
                    <property name="skill"></property>          </subclass>
          <subclass name="Sales" <strong><span style="color:
                     #ff0000;">discriminator-value="2"</span></strong>>
                    <join table="sales">
               <key column="employee_id"></key>
               <property name="sell"></property>
                    </join>
          </joined-subclass>
     </class>
</hibernate-mapping>
```

For the aggregation, the system is an aggregation of components. In UML, aggregation is shown by an open diamond on the end of the association line that points to the parent (aggregated) class. There is an implied multiplicity on this end of 0...1, with multiplicity of the other end shown in the diagram as usual. To describe this association, we would say that each system is composed of one or more components and each component is part of zero or one system. Since the component can exist by itself in this model (without being part of a system), the system name cannot be a part of its PK. We'll use the only candidate key {type, mfgr, model, SN} as PK, since this class is not a parent. The system name, just an FK here, will be filled in if this component is installed as part of a system; it will be null otherwise. In UML, there is a stronger form of aggregation that is called composition. The notation is similar, using a filled-in diamond instead of an open one. In composition, component instances cannot exist on their own without a parent; they are created with (or after) the parent and they are deleted if the parent is deleted. The implied multiplicity on the "diamond" end of the association is therefore 1..1.s

For the recursive association, for example, the class Employee has recursive association itself. Normally, we wouldn't show a foreign key in the class diagram; however, including the manager as an attribute of the employee here (in addition to the association line) can help in understanding the model. In the relation scheme, we can explicitly show the connection between the surrogate primary key (employeeID) and the managerID (which is a foreign key, even though it is in the same scheme).

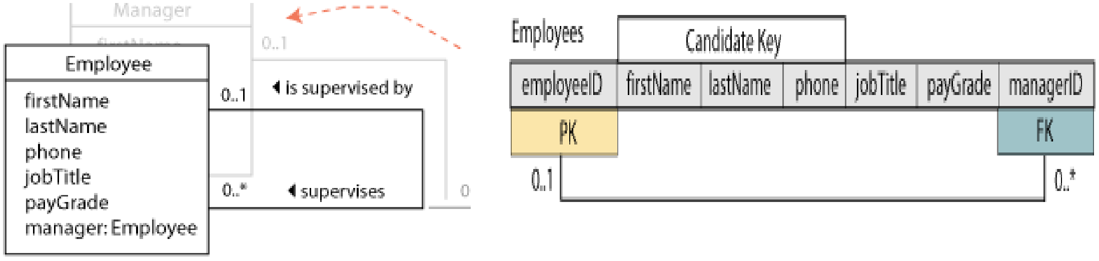


Figure 4-4 The example of recursive association

1.1.1.6 4.2.2.3 Hibernate mapping files

Hibernate configuration files, and XML mapping metadata files need writing. . "com.mysql.jdbc.Driver" is used to set database connection driver. "jdbc:mysql://localhost:3306/careplandb" is used to connect to the local database and the name of database is the careplandb. The label "<mapping

resource="PlandeSoinsPrevisionel/Acte.hbm.xml"/>” is to map hibernate mapping file.

```
<hibernate-configuration>
    <session-factory>
        <property
name="hibernate.connection.driver_class">com.mysql.jdbc.Driver</property>
   <property
name="hibernate.connection.url">jdbc:mysql://localhost:3306/careplandb</property>
        <property name="hibernate.connection.username">root</property>
        <property name="hibernate.connection.password"></property>
        <property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>
        <mapping resource="PlandeSoinsPrevisionel/Acte.hbm.xml"/>
    </session-factory>
</hibernate-configuration>
```

1.1.1.7 4.2.2.4 Hibernate mapping type

Hibernate will translate Java types to SQL / database types for the properties of your mapped classes. The table 4-1 shows that java type corresponds to hibernate type and sql type.

4.3 Implementation of Graphical user interface layer

The graphical user interface as an intelligent tool helps coordinators to make home care plan and save the information into the database. As a part of the project Plans’O’Soins, it can supply the accurate information for the other group of this project.

Table 4-1 Hibernate mapping types

Java type	Hibernate type	SQL type
java.lang.String	string	VARCHAR
java.util.Date	date, time	DATE, TIME
java.lang.Integer, int	integer	INT
java.lang.Class	class	VARCHAR
ava.io.Serializable	serializable	BLOB, BINARY

The user interface follows the whole process to specify the home care plans for each patient. Because it will be used for French local hostipal or home care center, this graphical user interface will be designed and implemented in English/French. Firstly, the user could choose the language that they are familiar with. The default language is French. Then the end user, namely the primary medical coordinator, has two options: querying the existing patient or create the new patient. By searching the patient, they will get the information from the database and all the information of patient will be displayed in the patient information panel. If the patient does not exist, the user can create a new record for this person. For the patient i information panel, the user could take some operations on it and finally save them into the database. For the existing patient, there has been a projected home care plan. They could modify them and store the modified home care plan as a new home care plan; For a new patient, the end user would specify the home care plan for them. Home care plan is constituted of an activity table which could be transformed to the intervention table and the agenda table so that the end user could classify the home care plan according to his requirement. After specifying the home care plan, all the data on home care plan would be stored into the database as the input of the automata.

The GUI Operation work flow as figure 4-5. Firstly, the user logs in the language option windows to select the language which they prefer. Then on the patient selected window, the coordinator could have two options: searching information of the existing patient and creating a piece new existing patient document. Patient information window will be used to show all the data of patient. After that, the coordinator choose activity table window to specify home care plan to this patient. For home care plan, it consists of a lot of activities, that is to say, finishing inputting activity table is the same as specifying home care plan. In the activity table, the most important task is specifying temporality for temporality table. Finally, user could generate intervention table automatically from activity table.

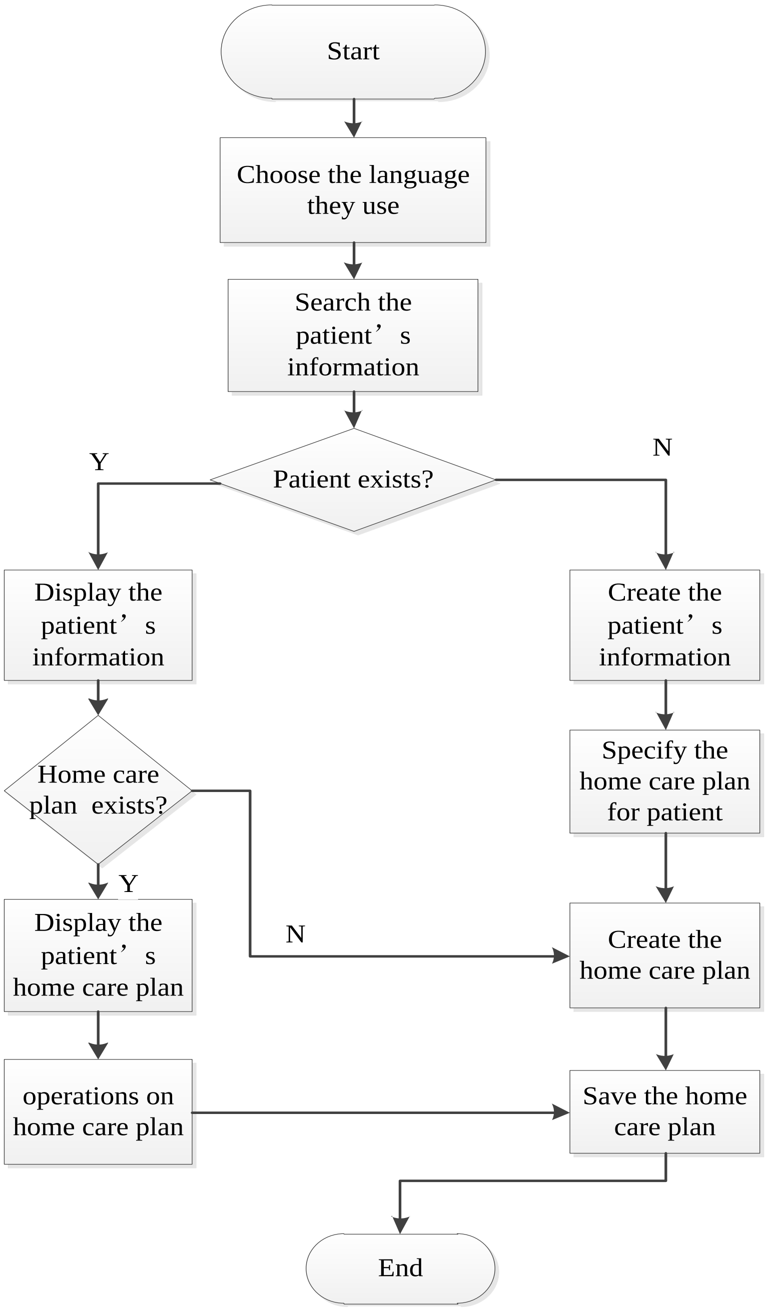
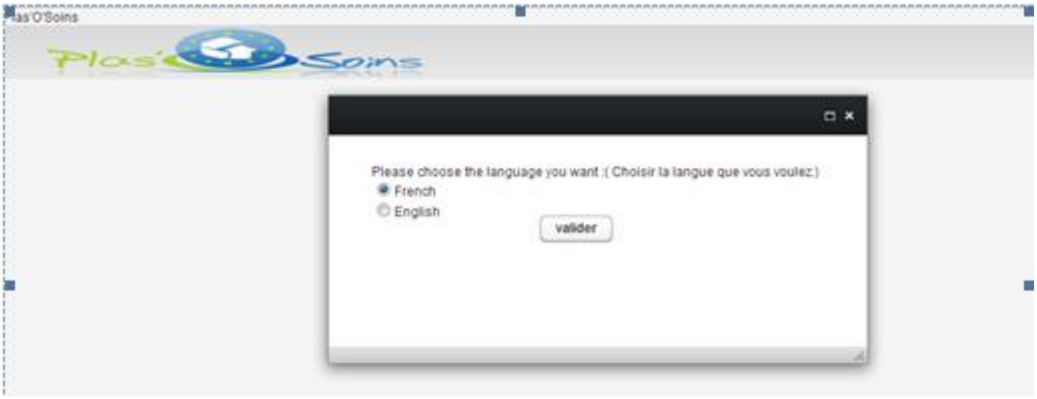

Start
Choose the language they use
Search the patient’s information
Patient exists?
Y
N
Display the patient’s information
Create the patient’s information
Home care plan exists?
Specify the home care plan for patient
Y
Display the patient’s home care plan
N
Create the home care plan
operations on home care plan
Save the home care plan
End

Figure 4-5 The work flow of graphical user interface

The language option window (see Figure 4-6) is displayed in the first page. On the top of page is the logo of the project. The default language is French because the medical coordinator works in the local hospital or medical care center. So I define a package constant, in the package, we define ConstantEnglish class and ConstantFrench class to save the information respectively.



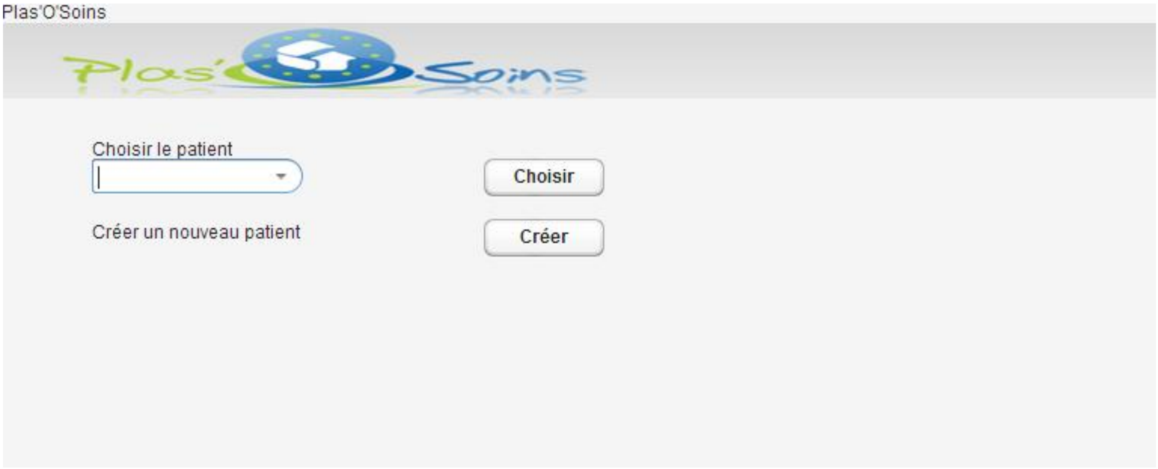
Plas'O'Soins

Please choose the language you want :(Choisir la langue que vous voulez:)

● French
○ English

Figure 4-6 Language selected window

After the coordinator finishes selecting the language, he can have two options: select one existing patient and create a new patient. The coordinator selects the existing patient by the droplist from the database. The window is displayed in the figure 4-7 where the coordinator is able to choose to create one record of patient information if the patient did not exist.



Plas'O'Soins

Choisir le patient

Créer un nouveau patient

Figure 4-7 The patient selected window

The patient information panel is shown as follows (Figure 4-7). The coordinator firstly input the first name and fimaly name for patient. For the date of birth, when it will be choosed, can be calculated to age directly. The weight and height are filled in to calculate to BMI (bady mass index) by the formulation $BMI = {}^{weight}\!/_{height^2}$. After that, telephone number and sex will be selected directly. For the address, it will be divided into details: name and number of the street, post code, city, building and department. The coordinator could follow the prompts of each textfield. The zone is selected from the database or input directly. Then the user could choose the treatment period and fill in his name as the modified person. In the database, modified time is the current time and all the date is in form of "MM/DD/YYYY".

And if the patient has information in the database, all the information will be displayed in the textfield. Each time when the user clicks the button 'valider', all the information will be stored into database. Others this panel is used to create a new record for a new patient.

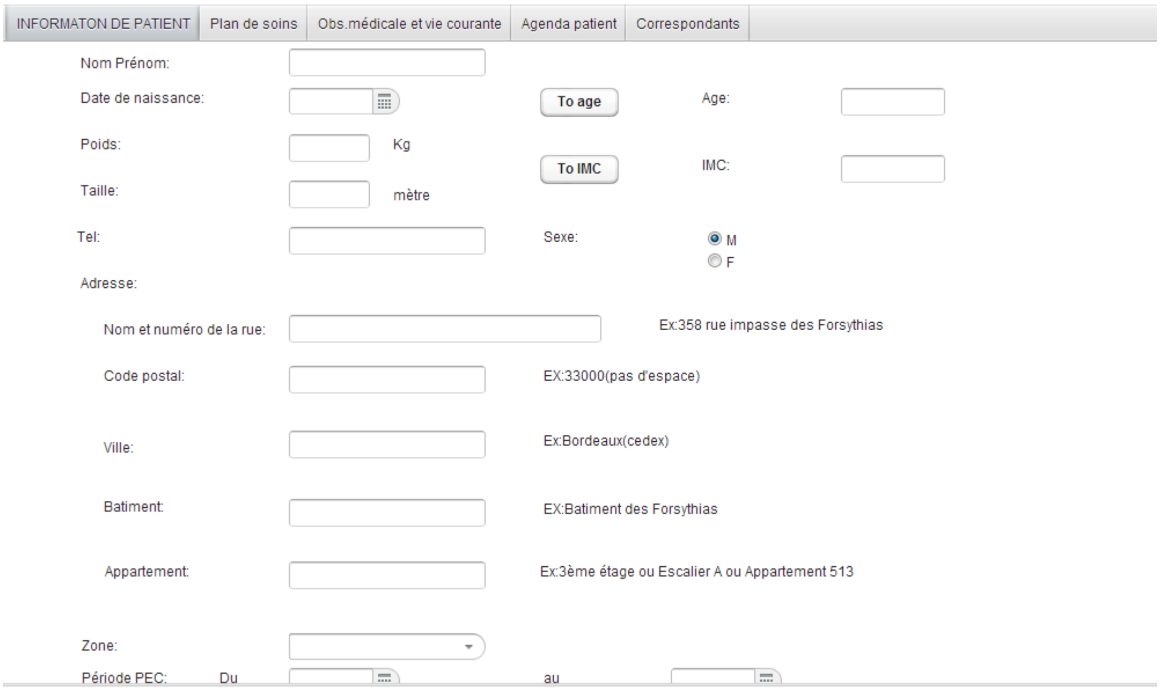
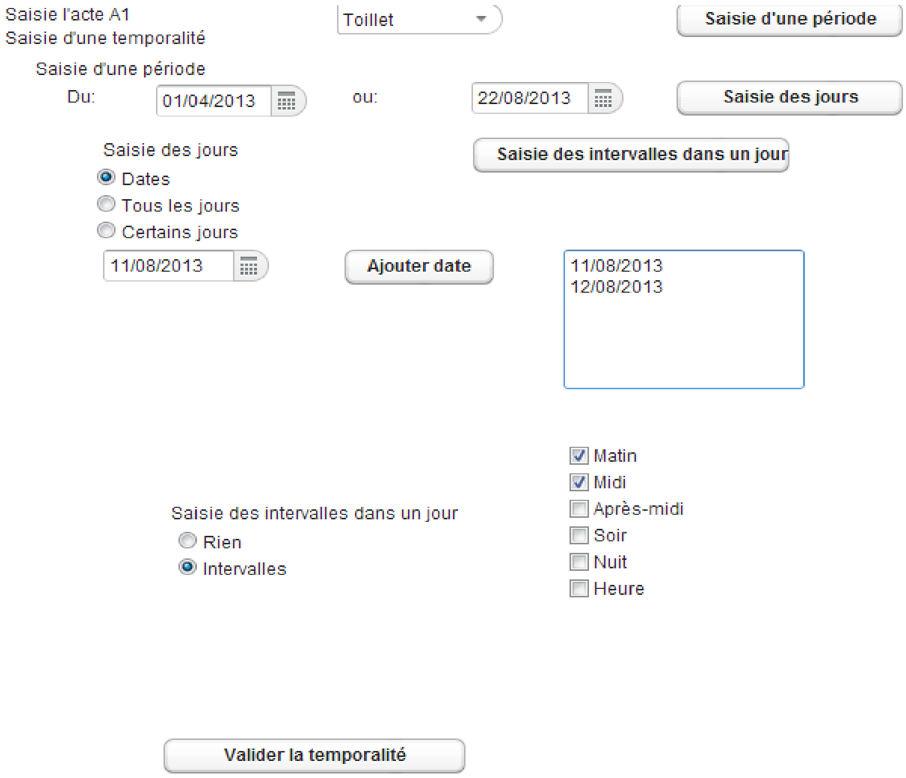


Figure 4-8 The patient information panel

The home care plan specifying panel consists of activity table and some buttons. Since the home care plan consist of different activities, filling in the activity table a main method to manange and specify home care plan. Firstly, click the button to specify the activity to home care plan. The temporality selected window was shown as figure 4-9. The user chooses the activity name for each activity, which is proposed by the system and the user could input directly. For the temporality, one activity could have more than one temporality. The default period of the home care plan is the totally same with the period of treatment. Meanwhile, the user could modify it during the period of treatment. After clicking the button 'choosing the days', there are four types of days: detailed dates, every day, some days and festivals. The user can use the data field component to add any dates which range is not beyond the period; Everyday means that all the dates during the whole period; Someday means that the user chooses some of them in one week from Monday to Sunday. After that, people can choose nothing or intervals as the time range. For the intervals, people can choose interval such as morning as well as detailed time such as 19h00. However, if detailed time is in the interval, for instance, the user chooses morning as interval and 09h00 as detailed time, there will be a conflict and the user interface will give the prompts to choose it again.



Saisie l'acte A1 Toillet

Saisie d'une temporalité

Saisie d'une période

Du: 01/04/2013 ou: 22/08/2013

Saisie des jours

● Dates

○ Tous les jours

○ Certains jours

11/08/2013

11/08/2013
12/08/2013

☑ Matin

☑ Midi

☐ Après-midi

☐ Soir

☐ Nuit

☐ Heure

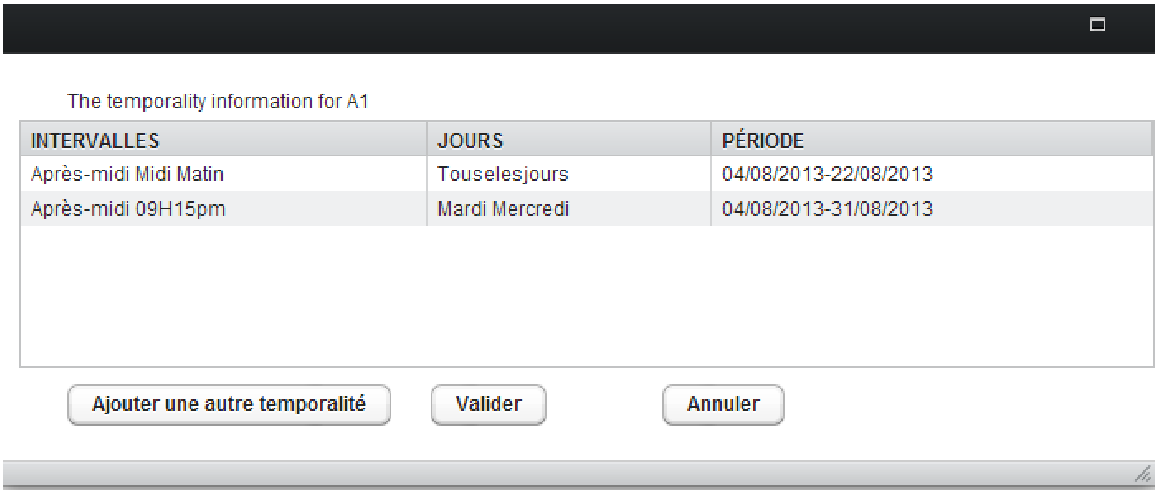
Saisie des intervalles dans un jour

○ Rien

● Intervalles

Figure 4-9 The patient information panel

After assigning the temporality for one activity, a temporality information table will be displayed to coordinators (see Figure 4-10). They can change temporality's order by droping and drugging one line of table. If the user wants to delete one of them, he only need drug it to the desktop. If the user wants to conitune to add another temporality to the table, he has to click the button "add another temporality". When the user does not modify it again, the content of temporality table will be displayed in the activity table.

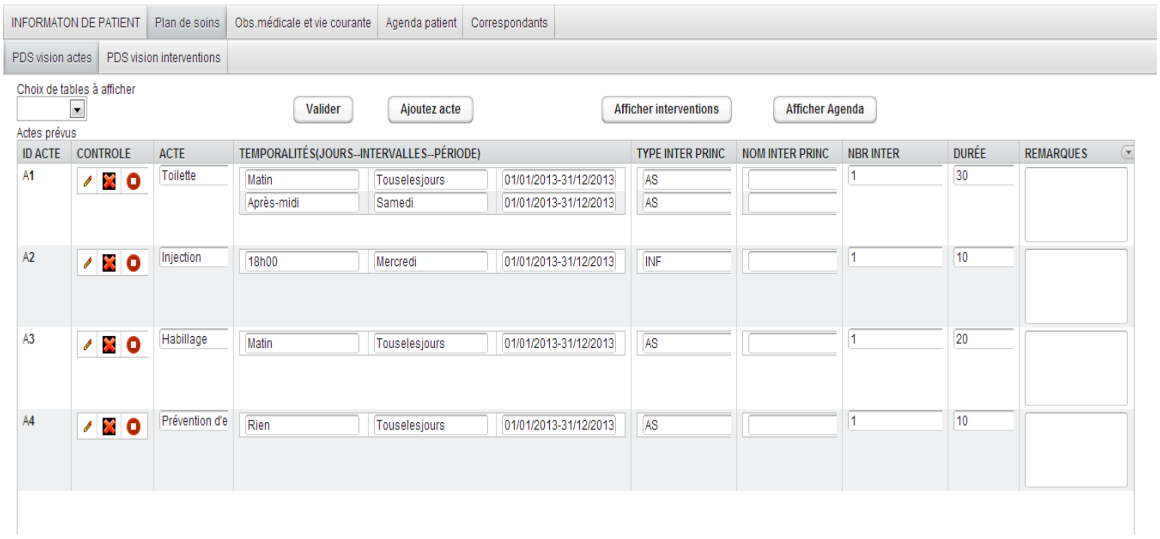


The temporality information for A1

INTERVALLES	JOURS	PÉRIODE
Après-midi Midi Matin	Touselesjours	04/08/2013-22/08/2013
Après-midi 09H15pm	Mardi Mercredi	04/08/2013-31/08/2013

Figure 4-10 The temporality table for an activity

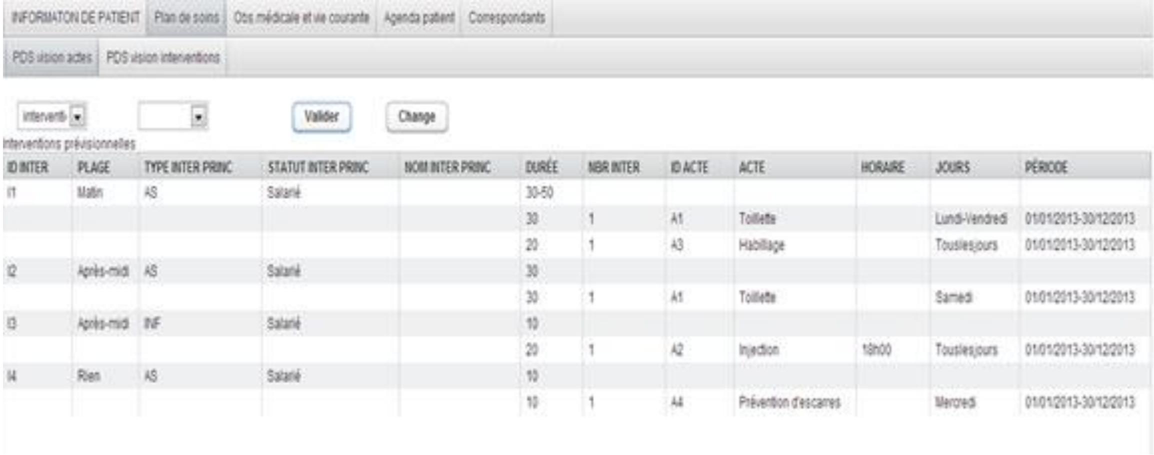
The activity table is shown in the Figure 4-11. For different activity, the ID is his unique identity. The control table has three icons to operate: modify (modify the attributes of the activity), delete (delete the activity directly but the ID was not changed) and stop (indicate that the activity is not feasible). For example, the activity A1 is described like this : its activity name: toilet; temporality1:01/01/2013-31/12/2013;everyday;moring,noon,afternoom; temporality2:01/01/2013-31/12/2013;Monday,Wednesday;09h15pm,afternoom; type of actor: AS; number of actor:1;duration:30 days; The user could comment on the activity.



ID ACTE	CONTROLE	ACTE	TEMPORALITÉS(JOURS--INTERVALLES--PÉRIODE)			TYPE INTER PRINC	NOM INTER PRINC	NBR INTER	DURÉE	REMARQUES
A1		Toilette	Matin Après-midi	Touselesjours Samedi	01/01/2013-31/12/2013 01/01/2013-31/12/2013	AS AS		1	30	
A2		Injection	18h00	Mercredi	01/01/2013-31/12/2013	INF		1	10	
A3		Habillage	Matin	Touselesjours	01/01/2013-31/12/2013	AS		1	20	
A4		Prévention d'e	Rien	Touselesjours	01/01/2013-31/12/2013	AS		1	10	

Figure 4-11 The activity table

After user clicked the button “generate intervention table”, the home care plan will be transformed to the type of intervention. There are two types of intervention table: the developed intervention table and the simplified intervention table. The user could choose the NativeSelect to view what he wants. In the developed intervention table, the home care plan was classified by days and type of actors. Therefore, for instance, for the intervention I_1, its constraints are “interval == morning||Type of intervention ==AS”. The intervention I_1 includes A_1 which duration is 30 days and A_3 which duration is 20 days. By the calculation, the duration of the intervention is 30-50days. For the activity table (Figure 4-11), the developed intervention table is shown as Figure 4-10. The home care plan has become four interventions: I_1 (interval:morning and type of actors: AS), I_2 (interval:afternoon and type of actors: AS), I_3 (interval: afternoon and type of actors: INF) and I_4 (interval:Rien and type of actors: AS). Then the information of each activity is shown in detail: number of actor, activity name, period and days. If there is detial time in the time ranges, it will be shown in the hour column.

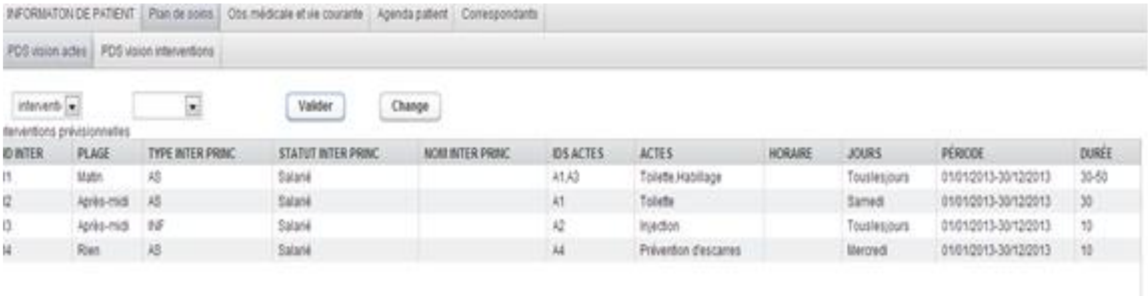


Interventions prévisionnelles

ID INTER	PLAGE	TYPE INTER PRINC	STATUT INTER PRINC	NOM INTER PRINC	DURÉE	NBR INTER	ID ACTE	ACTE	HORAIRE	JOURS	PÉRIODE
I1	Matin	AS	Salarié		30-50						
					30	1	A1	Toillette		Lundi-Vendredi	01/01/2013-30/12/2013
					20	1	A3	Habillage		Touslesjours	01/01/2013-30/12/2013
I2	Après-midi	AS	Salarié		30						
					30	1	A1	Toillette		Samedi	01/01/2013-30/12/2013
I3	Après-midi	INF	Salarié		10						
					20	1	A2	Injection	18h00	Touslesjours	01/01/2013-30/12/2013
I4	Rien	AS	Salarié		10						
					10	1	A4	Prévention d'escarres		Mercredi	01/01/2013-30/12/2013

Figure 4-12 The developed intervention table

Another table is the simplified intervention table. All the information is shown in one line. The identity of intervention, time range, type of actor and statute of actor are the same as the developed intervention table. The most difference is that it is the conclusion of the developed intervention table. For example, for the intervention I1, the corresponding activity is A1 toilet and A3 habillage. The end user could the columns that he would like to view. The new activity table which has one new column the ID of intervention will be generated by the button “submit”.



Interventions prévisionnelles

ID INTER	PLAGE	TYPE INTER PRINC	STATUT INTER PRINC	NOM INTER PRINC	IDS ACTES	ACTES	HORAIRE	JOURS	PÉRIODE	DURÉE
I1	Matin	AS	Salarié		A1,A3	Toilette,Habillage		Touslesjours	01/01/2013-30/12/2013	30-50
I2	Après-midi	AS	Salarié		A1	Toilette		Samedi	01/01/2013-30/12/2013	30
I3	Après-midi	INF	Salarié		A2	Injection		Touslesjours	01/01/2013-30/12/2013	10
I4	Rien	AS	Salarié		A4	Prévention d'escarres		Mercredi	01/01/2013-30/12/2013	10

Figure 4-13 The simplified intervention table

4.4 System Testing

The testing of system prototype focused on evaluating the novel DSL-based method and verifying the graphical user interface function from the end users. The database layer and the graphical user interface layer have been tested on execution-based testing and the logic layer has been tested on non-execution-base testing.

For the testing method, because we often met and communicated with the end user, we choose V-model testing method. The V-model deploys a well-structured method in which each phase can be implemented by the detailed documentation of the previous phase. Testing activities like test designing start at the beginning of the project well before coding and therefore saves a huge amount of the project time. The purpose of V model is to improve efficiency and effectiveness of software development and reflect the relationship between test activities and development activities as shown in Figure 4-14. V-model is perhaps the most traditional model followed for management of software tests.

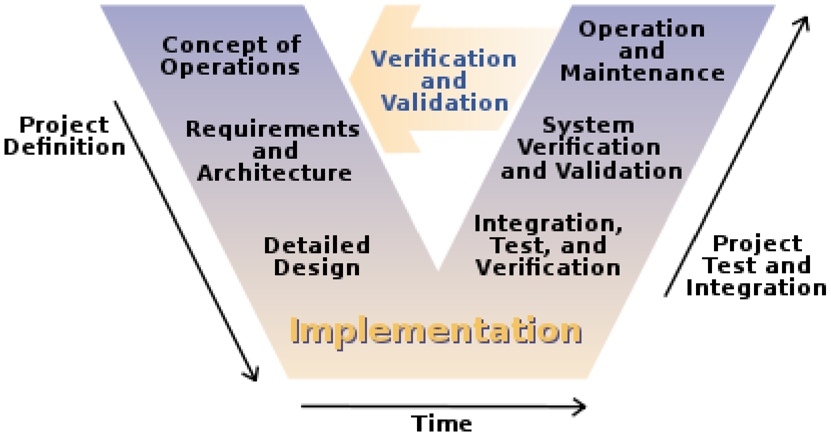

Concept of Operations
Verification and Validation
Operation and Maintenance
Project Definition
Requirements and Architecture
System Verification and Validation
Detailed Design
Integration, Test, and Verification
Project Test and Integration
Implementation
Time

Figure 4-14 V Model

4.4.1 Graphical user interface testing

Since the home care plan specifying process was described as the same as the operations on the graphical user interface testing, test cases were designed for each step to verify if our user interface could give out correct prompts or instructions for the primary medical coordinators. In the testing, a coordinator who is in charge of specifying home care plan would operate on the graphical user interface.

The testing of language option window focused on switching between two kinds of languages and testing the dual confirmation. The test cases are in 表 0-1.

Table 0-1 Test case language option

Test case ID	UI_Test_01
Test case description	The users choose language and test if it is successful or not.
Test requirement	None
Test steps	(1) The users choose the language they need and click the button "valider"; (2) The users did not choose any language and click the button "valider"
Predicted result	If click on French, all the information is described in French; If click on English, all the information is described in English; If does not chose, the prompt "choose the language";
Actual result	If click on French, all the information is described in French; If click on English, all the information is described in English; If does not chose, the prompt "choose the language";
Pass/fail	Pass

Table 0-2 Test case UI_Test_02

Test case ID	UI_Test_02
Test case description	The users choose the patient and verify the three situations
Test requirement	Successfully connection with the database
Test steps	(1) The coordinators select the patient (2) The coordinators create a new patient
Predicted result	If the patient exists, all the information of patient will be displayed to the users. If the patient does not exist, the system give the prompt "No patient, please create it!" If the patient does exists, but his care plan is not complete, the user could add other home care plan into the database.
Actual result	If the patient exists, all the information of patient will be displayed to the users. If the patient does not exist, the system give the prompt "No

	patient, please create it!" If the patient does exists, but his care plan is not complete, the user could add other home care plan into the database.
Pass/fail	Pass

The testing of patient option window focused on three different situations on selecting the patient or creating a new patient. The test cases are in 表 0-1.

The testing of activity tables focused on the operations on activity panel, such as delete one row of activity table and modifies some data in the table. The test cases are in 表 0-1.

Table 0-3 Test case UI_Test_03

Test case ID	UI_Test_03
Test case description	Verify the accuracy of activity table and detect the operations on the activity table.
Test requirement	Successfully connection with the database.
Test steps	(1) Click the "add activity" to add one activity into the activity table (2) Check the information of activity table (3) Take operations on the activity table
Predicted result	The user could add activity successfully and each activity has its own ID. The information of activity table is accurate. Delete one of activities, and activity id does not be changed. Modify any attributes in the table, and database save it successfully
Actual result	The user could add activity successfully and each activity has its own ID. The information of activity table is accurate. Delete one of activities, and activity id does not be changed. Modify any attributes in the table, and database save it successfully
Pass/fail	Pass

The testing of temporality panel which is in the activity table focused on the process of specifying temporality for each activity. The test case is shown in Table 0-4.

Table 0-4 Test case UI_Test_04

Test case ID	UI_Test_04
Test case description	The temporality selected window is one step of adding an activity and verify the whole process.
Test requirement	Successfully connection with the database.
Test steps	(1) Choose the period (2) Choose the days (3) Choose the time ranges (4) The temporality table is shown to the user and the user could drag and drop the one of temporality (5) Save them into the activity table
Predicted result	(7) The default period will be provided by knowledge base (8) The user could modify the default period (9) 4 types of Days could be chose, especially the detailed date is not beyond the period (10) 2 types of time ranges could be selected. If the interval and detailed time has conflicts, the system will give the prompt. (11) The temporality table has drug and drop function (12) The information of temporality table is the same as that the temporality column of the activity table.
Actual result	(1) The default period will be provided by knowledge base (2) The user could modify the default period (3) 4 types of Days could be chose, especially the detailed date is not beyond the period (4) 2 types of time ranges could be selected. If the interval and detailed time has conflicts, the system will give the prompt. (5) The temporality table has drug and drop function (6) The information of temporality table is the same as that the temporality column of the activity table.

Pass/fail	Pass

The testing of intervention panel which is a new table focused on the transformation from activity table and intervention table. The test case is shown in Table 0-4.

Table 0-5 Test case UI_Test_05

Test case ID	UI_Test_05
Test case description	Verify the two kinds of intervention table
Test requirement	Successfully connection with the database.
Test steps	(1) Click the "generate intervention table" button (2) Choose which kind of intervention table generated (3) Verify the accuracy of intervention table compared to corresponding activity table.
Predicted result	(1) The intervention table will be generated successfully (2) Switch between two kinds of intervention table: developed intervention table and simplified intervention table (3) It is accurate that information of the intervention table
Actual result	(1) The intervention table will be generated successfully (2) Switch between two kinds of intervention table: developed intervention table and simplified intervention table (3) It is accurate that information of the intervention table
Pass/fail	Pass

4.4.2 Performance test for web's load

For the graphical user interface layer, we chose LoadRunner v9.0 as a testing tool which is a forecast system behavior and performance of industry standard load testing tools. LoadRunner is a kind of load testing tools that can forecast the behavior and properties of the system. LoadRunner find and make sure the problem, through the simulation of millions of users implement concurrent load and real-time performance monitoring. Through the use of LoadRunner, enterprise can shorten test time in maximum limit, optimize performance, and speed up the release cycle of the application system [42].

The LoadRunner test process as following

Step I: Planning the Test: successful load testing requires that you develop a thorough test plan. A clearly defined test plan will ensure that the LoadRunner scenarios that you develop will accomplish your load testing objectives. For our graphical user interface, our performance test case as follows (see Table 4-12):

Table 0-12 Performance test case

Test case ID	PT_Test_05
Test case description	Specify the home care plan
Test requirement	Successfully connection with the database.
Test steps	(1) Click the "add an activity" button (2) Follow the steps to specify the home care plans (3) After finishing specifying home care plans, click the "submit" button to store the information into the database (4) Click "generate the intervention table" button (5) View the intervention table (6) Logout the system
Numbers of user	30
Testing goal	Under transaction at the maximum number of online users, the client requests the average response time lese than XXs

Step II: Creating the Vuser scripts

Vusers emulate human users interacting with your client/server system. A Vuser script contains the actions that each virtual user performs during scenario execution. In each Vuser script you determine the tasks that will be: Performed by each Vuser, Performed simultaneously by multiple Vusers and Measured as transactions

Step III: Creating the Scenario

For each test case design test scenarios, the scene is set to reflect the performance of the system. LoadRunner v9.0, there are two types of scenarios, namely Manual Scenario and GoalOriented Scenario [7]. In a complex multi-choice test the former, because of its good flexibility. In which all the configurations require testers to configure it manually. Selected scenes will be added to the scene in the script, the script for each run is specified in the Quantity number of virtual users, execute the script tests the machine choose the machine can also be selected in the Add LoadGenerator other machines. Scene design is the core of Schedule settings; LoadRunner provides Schedule and Group two kinds of methods are Scenario. More rational simulate real operating conditions, usually choose Scenario, Run Mode Select Real sing life schedule, the number of users more choices Increasing pressure loading mode. A system based on the measured set loading specific manner. In addition, Web performance test system to simulate the general access to the browser, in the design scene, with each cycle requires a new user login system, empty cache contents, because the Web server each time the browser accesses static pages to really download the page, rather than from the cache access [8]. This setup ensures access to real user performance obtained when experience is better reflected in the performance test results.

Step IV: Running the Scenario

You emulate user load on the server by instructing multiple Vusers to perform tasks simultaneously. You can set the level of load by increasing and decreasing the number of Vusers that perform tasks at the same time.

Before you run a scenario, you set the scenario configuration. This determines how all the hosts and Vusers behave when you run the scenario. For each scenario, you create a list of scripts that Vusers run during scenario execution. You can run the entire scenario, individual Vusers, or groups of Vusers (Vuser Groups). While a scenario runs, LoadRunner measures and records the transactions that you defined in each Vuser script. The LoadRunner Testing Process

Step V: Analyzing Test Results

According to design of the use case for graphical user interface in a single test case which is responsible for specifying the home care plans, the number of concurrent users is 30. As a result, some of the test results reflect that the experience is better than the performance of the test results.

Table 0-12 Parts of test result of Performance test case

Script	Home.user
The total throughput	3215700
The total click number	960
Execution time	00:05:30
The average throughput	6648
The average click number	2.021

Analysis showed that with increasing number of virtual users, login transaction response time does not change significantly, indicating that the system can withstand 30 concurrent users login operation. In this scenario, the consumption of the resource map shows stable performance of the system to meet users' needs.

4.5 Brief summary

In this chapter, we have described the development environment of the project. And the implementations of three layers (database layer, user interface layer and logic layer) were explained in detail realized according to the design in chapter 3. The system testing, firstly, focused on the accuracy evaluation of the whole process of the transformation from meta-model to relational database. Then the graphical user interface will be tested following the whole process. The result has proved that the novel DSL-based method was accurate and reliable. Finally, home care plans were validated, and they met the requirements of the project.

Conclusion

This paper aimed at designing and implementing a novel based on domain-specific language method to manage the home care plan for the primary medical coordinator who is the end user in charge of specifying the home care plan. It firstly reviewed present techniques to manage the home care plan, such as business process modeling, however, some care plans can be viewed as a kind of non-structured business processes which, as a consequence, cannot be managed using standard BPM technology. To improve the efficiency and feasibility of managing home care plans, this paper suggested and addressed the problems underlying the design and management of home care plans. The proposed approach addresses the problem that a specific care plan is required for each specific patient and the obtained specification plays the role of a process model that must be enacted by a business process management or a workflow system. After analyzing the requirements from the end users and combining home care plan with Domain Specific Language based method, this research suggested the overall design. Considering the scalability, layered software architecture was proposed, including database layer, graphical user interface layer and logic layer. The database layer aims to create a robust relational database to provide the accurate data for graphical user interface layer and logic layer. The user interface layer provides an intelligent tool to specify home care plan and collect the information of patient and his corresponding home care plan. The temporality of home care plan as the input of timed automata model from logic layer were monitored by the model checker UPPAAL using temporality measurements through the analysis of model verification. The calibrated results were feed backed to detect the model and give the prompts to the end user. In terms of model analysis, we describe and discuss preliminary results regarding formalization of the proposed DSL abstractions using timed automata in order to provide basic services to support analysis, verification, enactment and management of home care plans. The system prototype demonstrated promising results in user testing. The results validated the novel DSL-based method to manage home care plan in terms of accuracy and reliability.

As an innovative new proposed method, the novel DSL-based approach can be further improved. Several future works are suggested. Firstly, our future research directions will be devoted to the development of the theoretical framework based on timed automata. Our preliminary results pave the way for a more detailed exploration of the benefit of using formal verification and model checking techniques in our context. Secondly, since the graphical user interface is the prototype of the final complicated version, other conditions like interface landscaping, touch screen functions will be introduced to improve interaction and user experience.

References

[43]R. HÉBERT, A. TOURIGNY et M.GAGNON. Intégrer les services pour le maintien de l'autonomie des personnes – PRISMA (Prog. de Recherche sur l'Intégration des Services pour le Maintien de l'Autonomie) - EDISEM – Canada – 2004.

[44]F. DUCHARM, P. LEBELE et H. BERGMAN. Vieillissement et soins, l'urgence d'offrir des services de santé intégrés aux familles du XX1ème siècle – Ruptures, revue transdisciplinaire en santé, vol. 8, n°2, p. 110-121 – 2001.

[45]NHS Gov.uk, "a care plan" (2012-2014), [Online] Available: http://www.homenursingagency.com (06 August 2012).

[46]P. Wohed, W. M. P. van der Aalst, M. Dumas, A. H. M. ter Hofstede and N. Russell. On the Suitability of BPMN for Business Process Modeling. Proceedings of the Fourth International Conference on Business Process Management (BPM 2006), volume 4102, pages 161-176, Vienna, Austria, September 2006. Springer.

[47]Mario Bochicchio, Anna Bruno*, Antonella Longo. Supporting Continuous Improvement in Care Management with BPM. Innovation Engineering Department, University of Salento, Software Engineering. 2011; pp: 32-38.

[48]M. Fowler and R. Parsons. Domain Specific Languages. Addison-Wesley, 2010, pp:15-18

[49]Browne, Eric Donald. Workflow Modeling of Coordinated Inter-Health-Provider Care Plans. Diss. University of South Australia, 2005, page: 12-14.

[50]J. Warren, G. Beliakov, J. Noone, and H. Frankel. Chronic disease coordinated care planning: flexible, task-centered decision support. In Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS-32), vol. 4. Maui, Hawaii, January 1999.

[51]Voutilainen, Päivi, Arja Isola, and Seija Muurinen. "Nursing documentation in nursing homes – state - of - the - art and implications for quality improvement."Scandinavian Journal of Caring Sciences 18.1 (2004): 72-81.

[52]Keller, W: Mapping Objects to tables, A pattern language, Proceedings EuroPlop (1997).

[53]Athanasiadis, I.N., Villa, F., Rizzalo, A.E.: Enabling knowledge-based software engineering through semantic-object-relational mappings. 3rd International Workshop on Semantic Web Enabled Software Engineering (2007).

[54]Grönroos, Marko. Book of Vaadin. Vaadin Limited, 2011, pp : 2-5

[55]Bauer, Christian, and Gavin King. Hibernate in action, 2005, pp : 21

[56]Rajeev Alur and David L. Dill. Automata for modeling real-time systems. In Proc. of Int. Colloquium on Algorithms, Languages, and Programming, volume 443 of LNCS, pages 322–335, 1990.

[57]John E. Hopcroft and Jeffrey D. Ullman. Introduction of Automata Theory, Languages, and Computation. Addison Wesley, 2001.

[58]Bengtsson, J., Larsen, K., Larsson, F., Pettersson, P., & Yi, W. (1996). UPPAAL—a tool suite for automatic verification of real-time systems (pp. 232-243). Springer Berlin Heidelberg.

[59]González, Oscar, Rubby Casallas, and Dirk Deridder. "Mmc-bpm: A domain-specific language for business processes analysis." Business Information Systems. Springer Berlin Heidelberg, 2009.

[60]Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel, D.: Semantic business process management: A vision towards using semantic web services for business process management. In: Lau, F.C.M., Lei, H., Meng, X., Wang, M. (eds.) ICEBE, pp. 535–540. IEEE Computer Society, Los Alamitos (2005)

[61]Miers, D., Harmon, P., Hall, C.: The 2007 bpm suites report – a detailed analysis of bpm suites version 2.1, Business Process Trends (2007), http://www.bptrends.com/

[62]Lau, C., Peddle, S., Yang, S.: Gathering monitoring metrics to analyze your business process, IBM (December 2007), http://www.ibm.com/us/

[63]P. Dadam, M. Reichertand, and K. Kuhn. Clinical workows - the killer application for process-oriented information systems Business, pages 1{15, 2000.

[64]R. Alur. Timed automata. In N. Halbwachs and D. A. Peled, editors, Proceedings of the 11th International Conference on Computer Aided Verification (CAV’99), volume 1633 of Lecture Notes in Computer Science.

[65]J. Bau, E. Bursztein, D. Gupta, and J. C. Mitchell, "State of the Art: Automated Black-Box Web Application Vulnerability Testing," in IEEE Symposium on Security and Privacy, 2010, pp. 332-345.

[66]Jesse James Garrett. (2005) Adaptive Path. [Online]. http://www.adaptivepath.com/publications/essays/archives/000385.php

[67]K. Benjamin, G.v Bochmann, G.V Jourdan, and I.V Onut, "Some Modeling Challenges when Testing Rich Internet Applications for Security," in First International workshop on modeling and detection of vulnerabilities (MDV 2010), Paris, 2010.

[68]A. Arasu, J. Cho, A. Garcia-Molina, A Paepcke, and S Raghavan, "Searching the web," ACM Transactions on Internet Technology, vol. 1(1), pp. 2-43, 2001.

[69]S. Brin and L. Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine," Computer Networks and ISDN Systems, vol. 30(1-7), pp. 107-117, 1998.

[70]Cirilo C E, do Prado A F, de Souza W L, et al. Building Adaptive Rich Interfaces for Interactive Ubiquitous Applications [J].

[71]Fowler M. Domain-specific languages [M]. Pearson Education, 2010.

[72]D. Amalfitano, R. Fasolino, and P. Tramontana, "Rich Internet Application Testing Using Execution Trace Data," in Proceddings of Third International Conference on Software Testing, Verification, and Validation Workshops , Washington, DC, USA, 2010, pp. 274-283.

[73]M. Bouet, K. Gani, M. Schneider, and F. Toumani. A general model for specifying near periodic recurrent activities - application to home care activities. Technical report-LIMOS, 2013.

[74]Kamara Benjamin, Gregor von Bochmann, Mustafa Emre Dincturk, Guy-Vincent Jourdan, and Iosif Viorel Onut, "A Strategy for Efficient Crawling of Rich Internet Applications," in Web Engineering: 11th International Conference, ICWE 2011,Paphos, Cyprus, Sören Auer, Oscar Díaz, and George Papadopoulos, Eds.: Springer Berlin / Heidelberg, 2011, vol. 6757, pp. 74-89.

[75]Antonsson M. Modeling of real-time systems in UML with Rational Rose and Rose Real-Time based on RUP [J]. 2001.

[76]Fern ́andez-Luque, F., Zapata, J., Ruiz, R., Iborra, E.: A wireless sensor network for assisted living at home of elderly people. Lecture Notes in Computer Science 5602 (2009) 65–74

[77]Dong J S, Hao P, Qin S, et al. Timed automata patterns [J]. Software Engineering, IEEE Transactions on, 2008, 34(6): 844-859.

[78]Bengtsson, J., Yi, W.: Timed automata: Semantics, algorithms and tools. In:

Lectures on Concurrency and Petri Nets. (2003) 87–124

[79]Tripakis, S., Dang, T.: Modeling, verification and testing using timed and hybrid automata. In Mosterman, P., Nicolescu, G., eds.: Model-Based Design for Embedded Systems, CRC Press (2009) 383–436

[80]Bot´ıa, J.A., Villa, A., Palma, J.T.: Ambient assisted living system for in-home monitoring of healthy independent elders. Expert Syst. Appl. 39(9) (2012) 8136–8148

[81]Lientz B P, Swanson E B. Software maintenance management: a study of the maintenance of computer application software in 487 data processing organizations [J]. 1980.

[82]Morris, John N., et al. "Comprehensive clinical assessment in community setting: applicability of the MDS-HC." Journal of the American Geriatrics Society 45.8 (1997): 1017-1024.

[83]Kempe, Allison, et al. "Quality of care and use of the medical home in a state-funded capitated primary care plan for low-income children." Pediatrics 105.5 (2000): 1020-1028.

[84]Hui-li, Zhang, et al. "Research of Load Testing and Result Application Based on LoadRunner." (2012).

Statement of Originality and Letter of Authorization

学位论文原创性声明

Statement of Originality

本人郑重声明：此处所提交的学位论文《基于 DSL 的家庭护理计划管理方法及工具设计与实现 DESIGN AND IMPLEMENTATION OF A DSL-BASED APPROACH TO MANAGE HOME CARE PLANS》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果。且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：吴元哈　　　　日期：　2013 年 8 月 14 日

学位论文使用权限

Letter of Authorization

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：吴元哈　　　　日期：　2013 年 8 月 14 日

导师签名：　　　　日期：　2013 年　11 月　26 日

Acknowledgement

I would like to express my deep gratitude to my master thesis advisor, Professor Farouk Toumani. Thanks him for offering me the internship and guiding me during the internship, for his patience, motivation, enthusiasm, and immense knowledge. I have learned many things since I became Prof. Farouk Toumani's student. I am also grateful to Professor Kun-Mean Hou from ISIMA and Professor Lanshun Nie from HIT for spending time reading my thesis and providing useful suggestions about this thesis. Their guidance helped me in all the time of research and writing of this thesis.

Special thanks are given to School of Software of Harbin Institute of Technology and Institut Supérieur d'Informatique de Modélisation et de leurs Applications. In addition, I want to thank all the professors for teaching us wholeheartedly.

I also want to thank every member of our project team in LIMOS, especially Gani Kahina, Sneijder Michel and M. Bouet, for their help. They are both my friends and professors. Without their guidance, I cannot finish my project at all.

Resume

Yuanling WU

19, Rue Roche Genès - 63173 Aubière Cédex, France
+33(0)641674992
wuyuanling555@163.com

Profile

An enthusiastic and dynamic student specializes in business process modeling software development and database research, and has more than 2 year’s internship experience of website and application. I am quick to assimilate new ideas, concepts as well as cutting-edge technologies and able to work well in a team. I am seeking a new role with scope for technical challenge and career progression.

Education Experience

2011.9-Present Dual degree: Engineer’s degree Bac+5 (France), Master’s degree (China) – Computer Science, Software Engineering - Institut Supérieur d'Informatique de Modélisation et de leurs Applications (ISIMA), Harbin Institute of Thechnology (HIT)

2007.9-2011.7 Bachelor - Software Engineering – Harbin Institute of Technology China

Project Experience

- **A DSL-based approach to manage home care plans (Internship)**

Le LIMOS (Laboratoire d’Informatique, de Modélisation et d’Optimisation des Systèmes)

April 2013 – present | Clermont-Ferrand, France

Description: Designing and implementing a novel DSL-based approach to manage home care plans. This new method will replace the traditional method and improve the efficiency of specifying the home care plans

- **peer code review management system (PCR), responsible for source-level software quality assurance and student's course management (Internship)**

Lab of software engineering school HIT (Harbin Institute of Technology)

October 2010 – June 2011 (10 months)

Description: Designing and implementing the system for code review management to assure the code quality and efficiency of code review

- **sina micro-blogging application, responsible for data collection, data mining and social network (Internship)**

Lab of software engineering school HIT (Harbin Institute of Technology)
October 2011 – January 2012 (4 months)

Description: Designing and implementing sina micro-blogging application to find social network by data mining.

Personal qualities

I have strong communication skill and management ability, and good at interpersonal relationship. As a quick learner and hard worker, I prefer all kinds of challenges and difficulties from techniques and life and have the courage to solve these problems. I believe that change favors the prepared mind.

Technical Skills

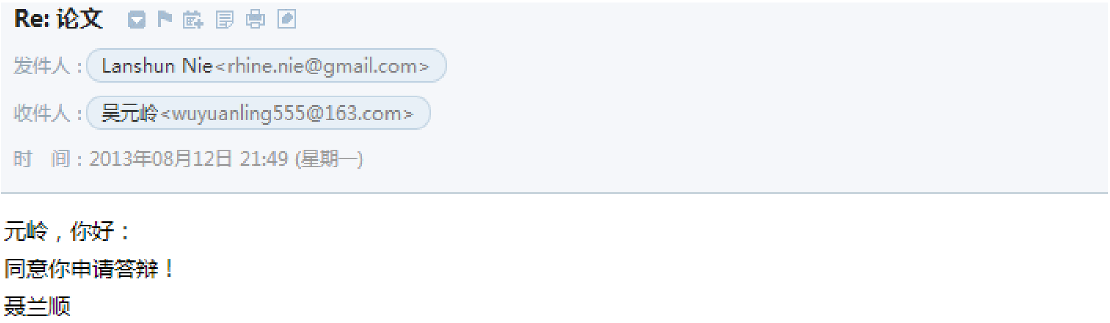
- 2+ years of ***project leadership*** experience and 4+ years of ***commercial software design and development*** experience.
- Advanced knowledge on ***Business Process Modeling*** fields.
- 4+ years of OO, multi-threaded, software development experience utilizing ***.NET(C#) and Java***.
- Extensive Internet based software development experience using ***ASP.NET, Eclipse (J2EE) and JavaScript (AJAX)***.
- Language: English (fluent, TOEIC: 770), Chinese (native), French (primary)

Core Courses Taken in School

Advanced Mathematical Analysis, Algebra and Geometry, Set Theory & Graph Theory, Probability Theory & Stochastic Process, Mathematics Modeling; C Language, Java, C++, .Net Framework; Computer Organization, Operating System, Principles of Compiler, Embedded System, Computer Networking, Graphical User Interface, Distributed Systems; Data Structure & Algorithms, Database, Software Component, Software Architecture, J2EE, Web Services; Software Engineering, Software Quality Assurance and Testing, System Analysis and Design.

Confirmation of Supervisors

HIT Supervisor



Re: 论文

发件人：Lanshun Nie<rhine.nie@gmail.com>

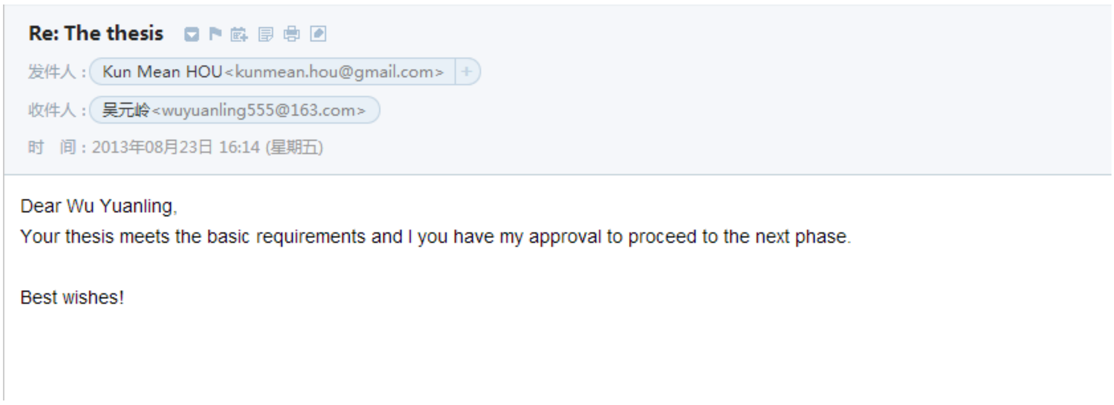
收件人：吴元岭<wuyuanling555@163.com>

时　间：2013年08月12日 21:49 (星期一)

元岭，你好：
同意你申请答辩！
聂兰顺

Signature: Date:

UBP Supervisor



Re: The thesis

发件人：Kun Mean HOU<kunmean.hou@gmail.com>

收件人：吴元岭<wuyuanling555@163.com>

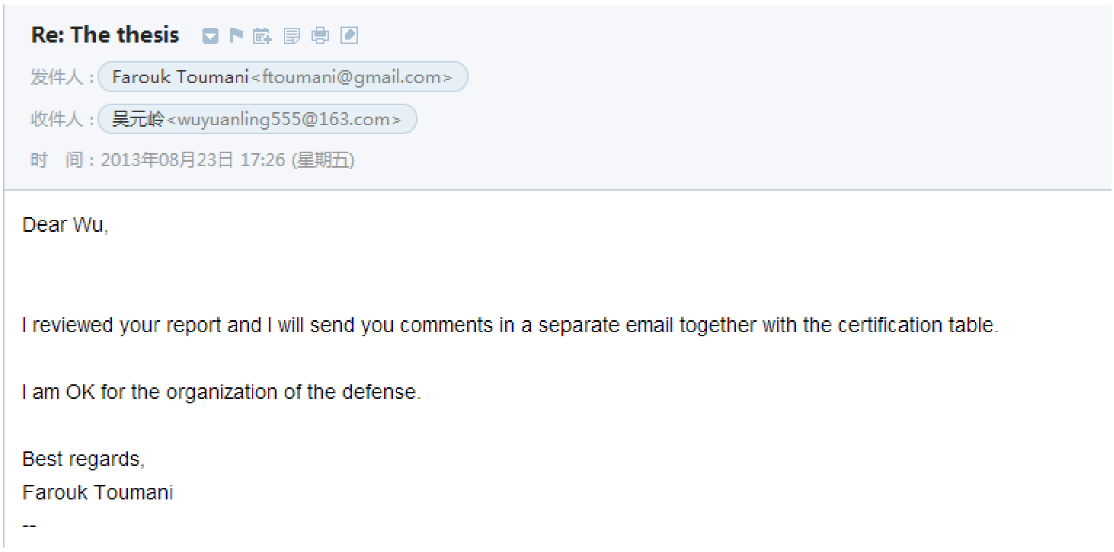
时　间：2013年08月23日 16:14 (星期五)

Dear Wu Yuanling,
Your thesis meets the basic requirements and I you have my approval to proceed to the next phase.

Best wishes!

Signature: Date:

Internship Supervisor



Re: The thesis

发件人：Farouk Toumani<ftoumani@gmail.com>

收件人：吴元岭<wuyuanling555@163.com>

时　间：2013年08月23日 17:26 (星期五)

Dear Wu,

I reviewed your report and I will send you comments in a separate email together with the certification table.

I am OK for the organization of the defense.

Best regards,
Farouk Toumani
--

Signature:　　　　　　　　Date:

基于DSL的家庭护理计划管理方法及工具设计与实现

作者： 吴元岭
学位授予单位： 哈尔滨工业大学