MASTER

A comparison of workflow management systems and clinical decision support systems in supporting clinical processes

Wang, X.

Award date:
2012

Eindhoven, November 2012

A Comparison of Workflow Management Systems and Clinical Decision Support Systems in Supporting Clinical Processes

by
Xuchen Wang

BSc Business Administration — INPG 2010
Student identity number 0757882

in partial fulfilment of the requirements for the degree of

Master of Science
in Operations Management and Logistics

Supervisors:
Prof.dr.ir. U.Kaymak, TU/e, Information Systems
dr.ir. H.A.Reijers, TU/e, Information Systems

TUE. School of Industrial Engineering.
Series Master Theses Operations Management and Logistics

Subject headings: Workflow, Clinical Support

i. Preface and acknowledgements

Writing these words marks the end of my master life. The past two years of studying and living in a foreign country are invaluable experiences for me. There are sorrow and happiness, loss and achievement, frustration and cheered-up. As my favorite quote says, what cannot defeat you will make you stronger, until I successfully finish the master thesis, I know that I finally make myself stronger. But my completion of study won't succeed if without the help and support from those people.

I first would like to thank my mentor and first supervisor, Professor Uzay Kaymak, for educating me along the hardest time of my master. I appreciate his understanding, help and support to my master project. Second I would like to give my many thanks to my second supervisor, Professor Hajo Reijers, who has not only spared effort on instructing my master thesis but also provided me with a lot of useful advices for my studies.

Next I would like to thank Dr. Paul de Clercq, Professor Erik Korsten, Mr. Harald van den Meerendonk, Ms. Corry Smeulders and PHD student Shan Nan who support my project and generously shared their knowledge with me throughout the project. Without their help and input of knowledge, I wouldn't have finished the project as smoothly as now I have.

Special thanks to my family and friends. My family is always the biggest supporter of my entire life and their trust is always the incentive for me to work harder. The great support from my boyfriend encourages me to go through the hardest moment of my study life and share my pains and gains. My best friends are always there to cheer me up whenever I felt down and to give me comforts.

Many thanks to many other people who have helped me during my two years' master life. Finally I'm finishing my two-year master study, and as an end to my study life in the past 24 years and a start for my career, I'll never regret to have this experience and to meet those people in my life. To all those people, thank you for making a better me.

Xuchen Wang

汪栩辰

November, 2012

Eindhoven

ii. Abstract

Healthcare institutions are facing the challenge to deliver safe care against affordable costs. Both workflow management systems and clinical decision support systems are powerful tools in enhance clinical services but from two different fields. This projects aims to investigate the commonalities and differences of applying workflow management systems (WFMSs) and clinical decision support systems (CDSSs) in clinical environment. By conducting theoretical analyses and practical implementations, the functionalities and the advantage and disadvantages of both systems will be discussed. In the end of the project, the proposal of combining the two types of systems to achieve better clinical care is put forward.

iii. Executive Summary

A Workflow management system (WFMS) is defined by the Workflow Management Coalition as “A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and, where required, invoke the use of IT tools and applications”("Workflow Management Coalition," 2012). The objective of a workflow management system is to handle these cases (by executing tasks) as efficiently and effectively as possible. The Clinical Decision Support System (CDSS) is an interactive decision support system that is designed to assist physicians and other health professionals with clinical decision-making tasks. It is defined by (Wyatt & Spiegelhalter, 1991) as “an active knowledge system which uses two or more items of patient data to generate case-specific advice”. Clinical decision support systems (CDSSs) are broadly classified into two main groups: knowledge-based CDSSs and Non-knowledge-based CDSSs (Berner & Lande, 2007).

Due to the increase of demand for healthcare, increasing pressure is being put on hospitals to work in the most efficient way thus to provide high quality and safe services while reducing costs. As a matter of fact, both the Medical Informatics community and business process management (BPM) community have spared effort on tackling the challenges. The Medical Informatics community mainly focuses on supporting patient care processes by developing electronic medical record (EMR) and computer-interpretable guideline (CIG) for clinical decision support systems while BPM mainly targets at enabling IT support for clinical processes based on workflow technology (Lenz, Peleg, & Reichert, 2012).

Yet, there are no researches that focus on analyzing in detail about how different the two systems are; also seldom researches mention the common functioning fields of both the two communities can contribute to. In other words, the distinct and overlapped functions of both workflow management systems and clinical decision support systems are still not clear in current researches. Here comes the goal of this project: to investigate the function of workflow management systems (WFMSs) and guideline-based decision support system in clinical environment, the advantage and disadvantages of both systems, whether they can replace each other in some field, and the possible improvement of the two types of systems. Based on the literature studies with respect to workflow systems and clinical guidelines, the main research question can be derived:

Main research question: How do Workflow Management Systems and Clinical Decision Support Systems differ from each other in supporting clinical processes?

The main hypothesis of the project is formulated after the main research question:

Main hypothesis: The functions of WFMSs and CDSSs are partly overlapping; they both have their own properties and features that cannot be replaced by each other. The main hypothesis is in accord with (Figure 3)**.**

In the project, both theoretical analyses and practical implementations are conducted. Specifically, three flexible workflow management systems (YAWL, FLOWer and Declare) and one knowledge-based clinical decision support system (Gaston) are chosen as the comparing

systems in the project. In the theoretical analyses, four scopes with eight dimensions are compared between YAWL, FLOWer, Declare and Gaston. Part of the analyses results can be presented in the table 1 below. Furthermore, the lifecycle of using the two types of systems are compared.

Scopes	Dimensions	YAWL	FLOWer	Declare	Gaston
Domain Scope	**Application goal**	Coordinate tasks, improve efficiency	Coordinate tasks, improve efficiency	Coordinate tasks, improve efficiency	Remind clinicians of tasks, improve security of clinical decisions
	Application independency	Domain independence	Domain independence	Domain independence	Domain independence
Technological Scope	**Type of support**	Workflow support, Process-centric	Workflow support, data-centric	Workflow support, Process-centric	Decision support, Patient/data-centric
Knowledge Translation Scope	**Guideline representation**	Process model	Process model	Process model	Decision tree flow chart
	Process definitions	Explicit process definition	Explicit/ implicit process definition	Implicit process definition	Explicit and implicit definition
	Modeling language	Workflow pattern	Case handling	Declarative workflow	Gaston representation model

Table i: Theoretical comparisons of the systems

In the practical implementation, the clinical guideline of anti-coagulation is executed in three systems: YAWL, Declare and Gaston. The practical implementations are evaluated based on observations of the executions and interviews with medical experts. For each system, discussions are made based on three dimensions: functionalities of the system, advantages and disadvantages and use perspective. The results of the practical implementations are depicted in table 2 below.

Systems	System functions	Advantages	Disadvantages	Use perspective
YAWL	Coordinate tasks, departments and people, enhance work efficiency	Reduce unnecessary communication time; Timed task to notify work to-do	Execution of task requires full knowledge of user; Time function insufficiency; Not sufficient to model every exceptional situations	Required basic knowledge of Unified Modeling Language (UML) and XQuery; Complex process configuration and data definition
Declare	Coordinate tasks and people while	Full support to the processes of	Data types are limited and simple;	Easier to start with; clinical experts can

	at the same time provide extremely flexible support	exceptional situations in clinical environment	manual execution for every task; processes are too much simplified by Declare; unfriendly user interface	build model by themselves in Declare
Gaston	Provide warnings to clinicians, notify critical issues	Facilitates the translation of paper-based clinical rules to executable models; track the condition of patient and give reminders	Manual trigger the start of Gaston; limited function to apply to other clinical fields	Easy to use: click a button then it will display information; modeling knowledge is required in building models in Gaston

Table ii: practical comparisons of the systems

Through theoretical analyses and practical implementation, the hypothesis of the project has been verified. In short, YAWL as workflow management system can support clinical rules to a moderate extent but this requires the clinicians to have full knowledge of the full content of the task in execution. But in the other way around, Gaston as clinical decision support system is not able to support organizational workflows due to the fact that it has no integration of resources. But in order to better support the anti-coagulation clinical guideline, it is proposed to integrate YAWL, Declare and Gaston, where Declare focus on coordination of task aggregations in high level, YAWL is responsible for modeling structured and repetitive processes in each task aggregations and Gaston provides decision support for the clinical rules in the process. But this possibility is required further examination, especially from a technical perspective.

iv. Table of Contents

v. List of Tables

vi. List of Figures

1. Introduction

This chapter is an introduction to the topic and research. First, the background information of the project is described, including relevant research, followed by the research question and methodology. Finally, the structure of the thesis is outlined.

1.1. Background Information

Information technology has been tremendously developed in the past twenty years and has been applied in various domains. In healthcare institutions, the application of health information technology has provided it with a great many benefits, like improved health care quality, reduced medical errors and health care costs, increased administrative efficiencies and so on. However, there are also challenges faced by healthcare institutions that call for tackling. On the one hand, increasing pressure is being put on hospitals to work in the most efficient way due to the increase of demand for healthcare; on the other hand, automating the clinical guideline execution process while at the same time ensuring the accuracy of clinical decisions is calling for further improvement.

Both Business Process Management (BPM) technology and Clinical Decision Support technology have endeavored to cope with the challenges. As one type of useful tools from the BPM field, Workflow Management Systems (WFMSs) support processes by managing the flow of work such that individual work items are done at the right time by the proper person. The benefits are that processes can be executed more rapidly and can be monitored(Mans, Van der Aalst, Russell, & Bakker, 2009). Computer Interpretable Guidelines (CIGs) as useful tools to standardize clinical guideline interpretation in Clinical Decision Support Systems (CDSSs) and thus the quality of clinical services offered by physicians can be improved. First, some relevant researches regarding those two fields are shortly introduced.

1.2. Relevant Researches

In terms of the categorization of workflows, a widely accepted taxonomy distinguishes between administrative, ad hoc, collaborative and production workflows (Alonso, Agrawal, El Abbadi, & Mohan, 1997). Based on that, researchers developed adaptive workflow, case-handling workflow that combines some characteristics of the traditional workflow types. In order to cater for the flexibilities in some processes, workflows types such as declarative workflow are further developed by researchers in order to meet the uncertain requirements or exceptional situations (Van der Aalst, Pesic, & Schonenberg, 2009). Clinical processes are of those flexible processes that differ from the workflows applied in productions and common organizations, the process management has to deal with more uncertainties and much complicated situations. To cater for the process flexibilities, WFMSs like ADEPT, FLOWer and Declare are dealing with adaptive, case-handling and declarative workflows respectively.

Clinical guidelines are useful tools to standardize and improve health care in support physicians making correct and safe clinical decisions. The Institute of Medicine (IOM) defined practice guidelines as ‘‘systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances'' ("Institute of Medicine," 2012). Modern researchers are putting more effort on developing guideline representation models to translate the paper-based clinical guidelines into computer interpretable guidelines (CIG) in an easy and standardized way, and on integrating the clinical decision support systems (CDSSs) with electronic medical records (EMRs) and the underlying information systems in the healthcare institutions(Lenz et al., 2012). Studies have shown that computer-based clinical decision support systems (CDSSs), when developed to provide patient-specific assistances in decision making and integrated with clinical workflow, can improve clinicians' compliance with Clinical Practice Guidelines (CPGs) and patient outcomes(Wang et al., 2002).

It seems that both researchers from BPM and Medical Informatics are working on the fields of their specialty since researchers have pointed out that both communities have addressed important aspects of supporting healthcare processes and the contributions of the communities seem to be quite complementary. Yet, there are no researches that mentioned the common functioning fields both the two communities could contribute to. In other words, it is uncertain whether workflow management systems and clinical decision support systems share some functionality in supporting healthcare processes.

1.3. Research Question and hypothesis

In a general point of view, workflow management systems and clinical decision support systems are likely to share some functionality in clinical decision support, as there are no previous studies that focus on this perspective, it would be interesting and meaningful to look into the idea. Especially from a practical point of view, this project can be helpful for the physicians in the healthcare institution to see how the two types of systems perform and to know more about their functioning in clinical decision support. It leads to the main objective of this project, which is to investigate the function of workflow management systems (WFMSs) and guideline-based decision support system in clinical environment, the advantage and disadvantages of both systems, whether they can replace each other in some field, and the possible improvement of the two types of systems.

Based on the literature studies with respect to workflow systems and clinical guidelines, the main research question can be derived:

- **Main research question**: How does Workflow Management System and Clinical Decision Support System differ from each other in supporting clinical processes?

To specify the main question and make it more concrete to understand, two sub-questions are developed:

- **Sub-question 1**: What are the commonalities and differences of the two systems?
  Motivation: Most researches study workflow management systems and clinical decision support systems in supporting organizational processes and clinical decisions respectively. Since there are seldom studies comparing the two systems in terms of their functionalities in clinical environment, it would be an interesting topic to firstly explore this question.
- **Sub-question 2**: what are the advantages and disadvantages of applying the systems?

Motivation: Current researches have studied workflow management systems and clinical decision support systems in terms of their own properties. However, seldom researches have compared the advantages and disadvantages of the two systems. Through conducting practical implementation of clinical guideline case in both types of systems, this perspective can be further explored.

In line with the research questions, four possibilities in terms of the relation between WFMSs and CDSSs can be drawn, as in the four figures below:

- **Possibility 1**: Workflow management systems (WFMSs) can fully perform as clinical decision support systems (CDSSs) in supporting clinical decision-making of physicians, furthermore, it has superiority and specialty that CDSSs are not able to equally perform (Figure 1);
- **Possibility 2**: WFMSs and CDSSs are equally comparable systems in clinical decision support, which means they expose same advantages and advantages in this field (Figure 2);
- **Possibility 3**:WFMSs and CDSSs share some functions in supporting physicians making decisions. However, they have their own advantages and disadvantages (Figure 3);
- **Possibility 4**: WFMSs cannot fully perform as CDSSs in supporting clinical decision-making of physicians; it can only perform limited functions of CDSSs and has insufficiencies in clinical decision support (Figure 4).

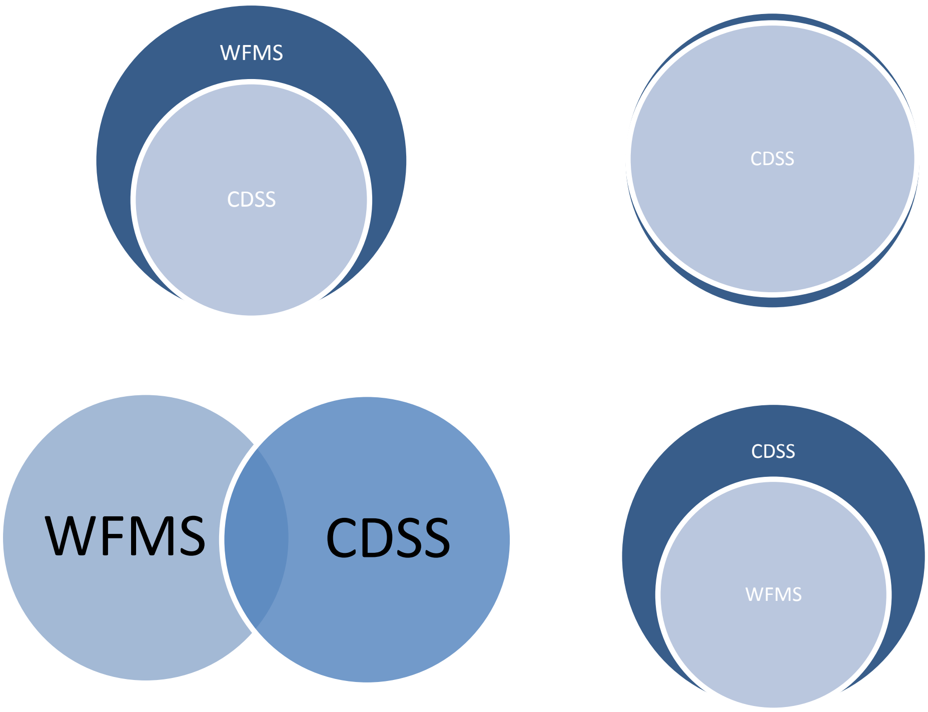


Figure 1, Figure 2, Figure 3, Figure 4: Relations of WFMSs and CDSSs

Based on the possibilities, the main hypothesis of the project can be formulated:

Main hypothesis: The functions of WFMSs and CDSSs are partly overlapping; they both have their own properties and features that cannot be replaced by each other, but some features are the same. The main hypothesis is in accord with (Figure 3).

1.4. Methodology

In this section, the research method that will be adopted in this project will be introduced, namely theoretical comparison and practical comparison.

1.4.1. Theoretical comparisons

In theoretical comparisons, one clinical decision support system with clinical guideline functionality will be chosen to compare with examples of three types of flexible workflow management systems. Moreover, knowledge of the clinical guideline will be thoroughly studied and analyzed. The steps involved in conducting the theoretical comparisons are:

- Collecting literatures with respect to workflow management systems, clinical guidelines and clinical decision support systems;
- Study and compare those literatures, summarize information that are related to characteristics of the systems and clinical guidelines;
- Write analyses based on the summaries;

After theoretical analysis of the three types of workflow management systems, two or all of them will be selected for practical comparison with clinical guideline cases.

1.4.2. Practical comparison

The practical comparison is to further verify the theoretical comparison result in previous section, as well as to re-examine the hypotheses proposed. The steps involved in the practical comparison are:

- Based on theoretical comparisons, decision will be made in terms of which workflow management systems are suitable for practical implementation and for further comparison.
- Given the research objective, clinical guideline case of anti-coagulation will be used in the comparison of practical implementation. The clinical guideline cases will be acquired from experts in Catherina Hospital Eindhoven.
- After the practical implementation of the clinical guideline cases in workflow management systems and one specific clinical decision support system, evaluations will be made base on experiences of the implementation.

1.5. Thesis Structure

In addition to this introduction chapter, the thesis consists of seven chapters. Chapter 2 is concerned with the state-of-the-art literatures that are relevant to this thesis topic. Specifically, the researches that are fulfilled and not fulfilled with respect to the topic, and the gaps between the researches are introduced. Besides, historical development of the systems, their applications in clinical environment is described. Chapter 3 is concerned with the methodology that is applied in this project. This chapter aims at answering three questions: what methodology is applied in this project, how the methodology is applied and why it is chosen for use. Chapter 4 depicts the background knowledge of certain types of workflow management systems, clinical decision support systems and the categorizations of clinical processes. Chapter 5 and Chapter 6 are the main chapter in the project. It describes both theoretical comparisons and practical implementations. After this, the analyses of both comparisons are conducted. Chapter 7 is the conclusion of the project which describes the main findings of the project followed by the limitations and future work.

2. An Overview of WFMSs and CDSSs

In this section, previous research literatures regarding workflow management systems and clinical decision support systems are thoroughly studied and described, specifically in terms of the system structure, historical development and current use in clinical environment. Moreover, the researches that are relevant to comparing those two fields are introduced, followed by a description of motivation for the main research question.

2.1. Workflow Management Systems and Clinical Decision Support Systems

A Workflow Management systems (WFMSs) and Clinical Decision Support Systems (CDSSs) are both the systems that have been widespread in modern life and made big contribution to the improvement of work efficiency and quality as well as the way of working. WFMSs are mostly adopted in common organizations to streamline the processes but rarely in healthcare institutions, while CDSSs are mainly used by healthcare institutions where clinical decisions are made and need support.

The Workflow management system (WFMS) is defined by the Workflow Management Coalition as "A system that defines, creates and manages the execution of workflows through the use of software, running on one or more workflow engines, which is able to interpret the process definition, interact with workflow participants and where required, invoke the use of IT tools and applications".("Workflow Management Coalition," 2012). The workflow reference model is one of the many principles used by the WFMC and is a general description of the architecture of a workflow management system. It describes the main components and the associated interfaces in the architecture (Figure 5)(Van der Aalst & Van Hee, 1997). The objective of a workflow management system is to handle these cases (by executing tasks) as efficiently and effectively as possible. The workflow process definition specifies which tasks need to be executed and in what order. Usually the execution of tasks is performed by one or more resources, e.g., a machine, an employee, etc.(Van der Aalst, Basten, Verbeek, Verkoulen, & Voorhoeve, 1999).

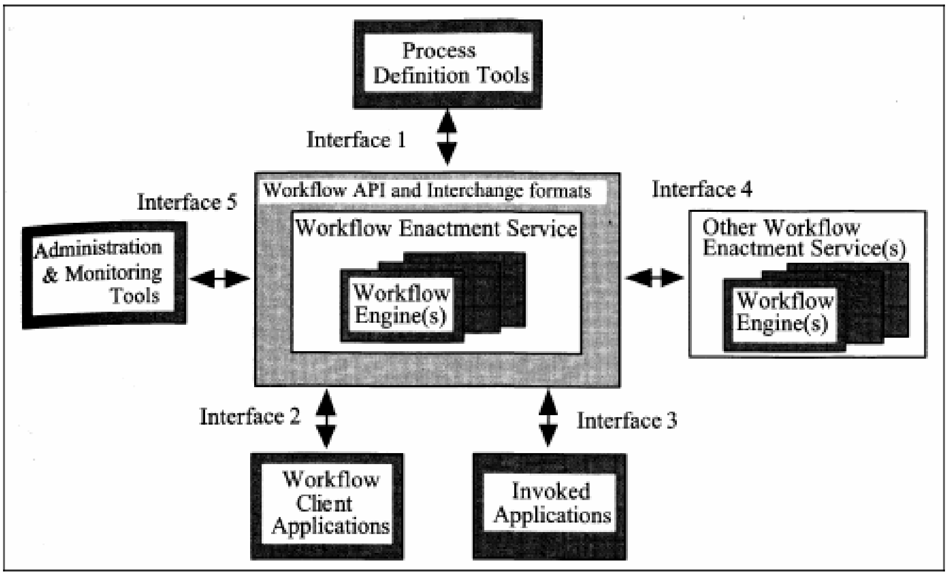


Figure 5: The Workflow Management Coalition's reference model

The Clinical Decision Support System (CDSS) is an interactive decision support system that is designed to assist physicians and other health professionals with clinical decision-making tasks. It is defined by (Wyatt & Spiegelhalter, 1991) as “an active knowledge system which uses two or more items of patient data to generate case-specific advice”. Modern methodology of utilizing CDSSs is to combine physicians’ knowledge and CDSSs to make a better analysis of patient data than either human or CDSSs could make by their own.

Clinical decision support systems (CDSSs) are broadly classified into two main groups: knowledge-based CDSSs and Non-knowledge-based CDSSs (Berner & Lande, 2007). The knowledge-based clinical decision support system contains rules mostly in the form of IF-THEN statements. It usually consists of three main parts: Knowledge base for containing the rules, Inference engine for combining rules with the patient data and the Mechanism to communicate for input and output of data. The knowledge-based CDSSs can be further categorized into rule-based systems and evidence-based systems two types. The non-knowledge-based CDSSs instead uses a form of artificial intelligence called machine learning; it can be further divided into Neural network and Genetic algorithms two categories.

2.1.1. System function and components

WFMSs coordinate procedural tasks and roles based on a pre-specified workflow model (Figure 6) (Cardoso, Bostrom, & Sheth, 2004) which is customized to accommodate specific business process structures. Once the design phase of the model is completed, instances are created to follow the actually steps in the given flow. During the execution, the instances can access legacy systems, databases, applications and interact with users. The workflow applications include a set of independent specifications describing how the task will be executed by whom and how the data will flow at runtime. A workflow management system usually requires several functional components:

1. Legacy system: The server is usually a powerful microcomputer, or a mini or mainframe computer.
2. Database: Database plays a major role in a workflow system. The workflow management system must be able to make use of a database management system that has already been chosen.
3. Applications: The applications support the performance of tasks, e.g. workflow editor.
4. Workflow executor: The workflow executor has to deal with each of the components listed above. It must be able to exchange information with the applications and the database system. Moreover, it must be able to cope efficiently with the available processing and network capacity.

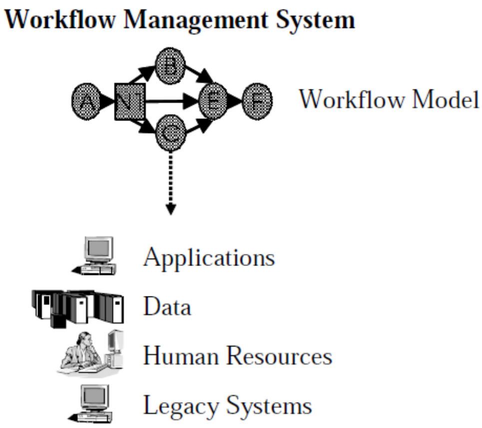


Figure 6: Overview of Workflow Management System

CDSSs are typically designed to integrate a medical knowledge base, patient data and an inference engine to generate case specific advice. The knowledge base contains the rules and associations of compiled data, which usually take the form of IF-THEN rules. Users could edit the knowledge base through an interface to keep the rules up-to-date. The inference engine combines the rules with patients' data.

The basic purpose of CDSSs is to help check routine clinical rules, thus to reduce the critical source of human error that won't be entirely eliminated with personal vigilance. Slightly more advanced systems can make analysis of clinical data and present guidance based on those data. Systems with more sophisticated functions can look for trends in value, thus to alert clinicians for important and prompt actions. The most complex systems attempt to aid clinical diagnosis but currently with quite limited implementation. Four key functions of electronic clinical decision support systems are outlined in (Perreault & Metzger, 1999):

1. Administrative: Supporting clinical coding and documentation, authorization of procedures, and referrals.
2. Managing clinical complexity and details: Keeping patients on research and chemotherapy protocols; tracking orders, referrals follow-up, and preventive care.
3. Cost control: Monitoring medication orders; avoiding duplicate or unnecessary tests.
4. Decision support: Supporting clinical diagnosis and treatment plan processes; and promoting the use of best practices, condition-specific guidelines, and population-based management.

2.1.2. Historical perspective

The historical evolution and maturity of WFMSs and CDSSs show two distinct movements. The study of these systems provides information valuable for understanding the actual difference between the two systems, which evolved from distinct domains.

The idea of workflow management systems arose in the 1990s. As the predecessor of workflow systems, the office information systems (OIS) field appeared earlier in the 1970s, when the first OIS prototypes were developed. Pioneer systems included the SCOOP project for automation of office procedures and Officetalk, which provided a visual electronic desktop metaphor, a set of personal productivity tools for manipulating information and a network environment for sharing information. When it comes to the 1980s, work was directed to the research of flexible groupware systems and models due to several failures of office automation projects and installations in OISs. In the 1990s, workflow management technology has emerged along with the reappearance of interest in OIS. New technological approaches, such as transaction process and integrated office systems provided fundamental support of technology for WFMSs. Research prototypes include METEOR, MOBILE, ADEPT, Exotica and MENTOR; commercial products include MQSeries Workflow, Staffware, TIBCO InConcert and COSA Workflow.

Historically, the origin of CDSSs can be traced back to the 1950s. The earliest days of computers developed. At that time, healthcare professional has put forward idea of using computers to assist them in the diagnostic process. The first article to discuss this possibility appeared in late 1950s (Ledley & Lusted, 1959), followed by experimental prototypes a few years later. However, problems occurred from limitations of scientific support to logistical difficulties have prevented widespread introduction of such systems. Not until the 1970s, three advisory systems have originated: deDombal's system for diagnosis of abdominal pain, Shortliffe's MYCIN system for selection of antibiotic therapy, and the HELP system for delivery of inpatient medical alerts. However, the early generation of CDSSs was not able to integrate with other computerized systems supporting hospital operation and patient data management, as a result, the advices generated by these CDSSs were often inaccurate or irrelevant, moreover, it functioned more like "decision-making" rather than "decision-support" which perceived by clinicians as a threat to their autonomy, therefore the early generation of CDSSs ended up as a failure in the 1990s. At the same time, with the significant technological breakthroughs such as the enterprise-level database management systems (DBMS) and health data standards like ICD and HL7, the use of computerized clinical information systems to support hospital operation as well as clinical activities started to flourish.

By analyzing the historical evolution and maturity of the two systems, we observe that both WFMSs and CDSSs emerged from a smooth, regular path at the beginning, encountered failure halfway, returned with new technology and finally proceed to a high point in the 1990s. In 2000s, research work focusing on integrating care flow support and clinical decision support appeared, which indicates both certain stability and the importance of such systems for organization, especially healthcare institution.

2.1.3. Applications in Clinical Environment

Historically speaking, business process support has been a major driver for enterprise information systems for a long time. Workflow technology has provided many benefits to organizational process management from a business point of view. The business-level benefits include better process control, improved efficiency and customer service, flexibility and business process improvement. Enterprise-wide, process-oriented information systems have been demanded by healthcare institutions for over 20 years. Dwivedi et al. first anticipated and explored the possibility of applying workflow management systems to managing repetitive patient administration procedures in healthcare institutions (Dwivedi, Bali, James, & Naguib, 2001). Workflow technology is currently applied in supporting care flow in healthcare institution in order to streamline the processes and reduce costs or to eliminate insufficient communication and missing information in the processes. Although workflow technology has delivered a great deal of productivity improvement, it has been mainly designed for the support of pre-defined and repetitive business processes requiring a basic level of coordination between human performers and some application services (Cardoso et al., 2004). In order to deal with the uncertainties, workflow paradigms like adaptive processes, case handling and declarative processes have emerged.

Besides the support of care flows in healthcare institutions, support of clinical decisions plays an equally important role in ensuring the correctness and efficiency of clinical services. Some clinical decision support systems apply medical knowledge by computer-interpretable guideline (CIG) formalisms to patient data for the purpose of generating case-specific decision support advice. It plays an instrumental role in helping health care achieve its ultimate goal, which is to provide high quality patient care while at the same time assuring patient safety and reducing costs. It is said to be the cornerstone of health informatics research and practice (Berner & Lande, 2007). Unlike general-purpose process models, the computer-interpretable guideline (CIG) formalisms focus more on the medical knowledge driving decision-making rather than on organizational issues such as scheduling and resource management.

By summarizing the application fields of both the systems, we can find that the focus of workflow management systems is different from that of clinical decision support systems: workflow management is process oriented, which means that it emphasizes the coordination of people and tasks, while clinical decision support is decision oriented, which means that it emphasizes the quality of clinical decision-making. It seems that from this functionality point of view, the two types of systems are distinct.

2.2. Gaps from literatures

Due to the increase of demand for healthcare, increasing pressure is being put on hospitals to work in the most efficient way thus to provide high quality and safe services while reducing costs. As a matter of fact, both the Medical Informatics community and business process management (BPM) community have spared the effort on tackling the challenges. The Medical Informatics community mainly focuses on supporting patient-care processes by developing electronic medical record (EMR) and computer-interpretable guideline (CIG) for clinical decision support systems while BPM mainly targets at enabling IT support for clinical processes based on workflow technology (Lenz et al., 2012).

Researchers from (Lenz et al., 2012) have also pointed out that both communities have addressed important aspects of supporting healthcare processes, and the contributions of the communities seem to be quite complementary. CIG formalism often does not address important managerial activities while BPM approaches often do not address complex decision criteria that include medical abstractions and temporal expressions. As suggested in (Terenziani, 2010), a hybrid approach could potentially be used for integrating approaches from both communities, where a computer-interpretable guideline approach is used to focus on “physician-oriented” issues, a workflow approach is used to cope with the related “business-oriented” issues, and the integration of them is obtained at the underlying semantic level. Yet, there are no researches that focus on analyzing in detail about how different the two systems are; also seldom researches mention the common functioning fields of both the two communities can contribute to. In other words, the distinct and overlapped functions of both workflow management systems and clinical decision support systems are still not clear in current researches. By studying and comparing the two types of systems, the characteristics and functionalities of both the systems can be understood better, thus the project might be helpful for the experts to further improve the systems and for users to choose a suitable one between the two systems based on their specific functions.

3. Evaluating Methodology

In order to develop a deeper understanding of both workflow management systems and clinical decision support systems in terms of their structure, characteristics, functions and etc., comparisons based on theoretical knowledge and practical implementation with real-life case are necessary. In this chapter, first the search strategy of finding relevant literatures for the theoretical comparison is introduced, then the research method that is adopted for comparing and evaluating the systems is described, followed by a description of how the method will be used as well as an explanation of why it is chosen.

3.1. Search strategy

Before working on the theoretical analyses, relevant articles and books are studied and compared. Several methods are applied for finding suitable literatures. The first one is to search by internet search engines; the second is a "snowball" method that starts from one article to find more articles that relate to it. Besides, some clinical institution websites are visited, from where several articles are mentioned and for further study.

Two search engines are used to find more scientific articles that are relevant to the thesis topic, namely TUE library search engine and Google Scholar. TUE library search engine is used only once for searching the papers written by dr. Paul de Clercq who is the inventor of one CDSS (Gaston) that will be looked into in this project. Google Scholar is more user friendly compared to TUE library search engine, it performs well in finding relevant articles with multiple key words and putting the most frequently cited articles in the beginning of the lists of search results. In some of the searches, the restriction to articles after 2001 has been applied in order to find latest study results. The searches were executed chronologically as listed in table 1, number of hits and selected articles are presented as well.

Search engine	**Search term used**	**Search restrictions**	**Number of hits**	**Selected articles**
TUE library search engine	Clercq	-	36	2
Google Scholar	Workflow management system	-	>100	4
Google Scholar	Workflow management history	-	>100	6
Google Scholar	Clinical decision support system	-	>100	4
Google Scholar	Clinical decision support system history	-	>100	2
Google Scholar	YAWL workflow system	From 2001	>100	2
Google Scholar	Declarative workflow system	From 2001	>100	2
Google Scholar	Case handling system	From 2001	>100	3
Google Scholar	Gaston clinical decision support system	-	>100	2

Table 1: Search engine and selected articles

The "snowball" method is applied in two ways. On one hand, it's used to find interesting articles that appear in the references of the relevant articles, while on the other hand, to find interesting articles that cite the relevant articles. In the former method, there's the pitfall that the articles in these reference lists are always older than the article itself, therefore the latter method is applied to find articles in this research field. The articles that are found by "Snowball method" are depicted in table 2.

From	**Method**	**Selected articles**
Lenz, Richard and Peleg, Mor and Reichert, Manfred (2012) Healthcare Process Support: Achievements, Challenges, Current Research. International Journal of Knowledge-Based Organizations (IJKBO)	From Reference list	9
P.A De Clercq, A Hasman, J.A Blom, H.H.M Korsten. Design and implementation of a framework to support the development of clinical guidelines. Int. J. Med. Inform., 64 (2001), pp. 285–318	From "Cited by" articles	1

Table 2: Snowball method

After finding the suitable literatures, information that related to the comparing dimensions is filtered and reflections on the filtered information are made into the analyses.

3.2. Theoretical analyses

In theoretical analyses, previous literatures regarding the three types of flexible workflow management systems (YAWL, FLOWer and Declare) and one type of clinical decision support system (Gaston) are thoroughly studied and compared. The three flexible WFMSs are chosen because they are the most popular and powerful flexible WFMSs. Gaston as a knowledge-based clinical decision support system is a quite successful CDSS that is applied in Catherina Hospital Eindhoven, also because it's the only CDSS that can be access to and obtained data, it's chosen as the only CDSS that will be studied in the project. The methodology of one specific article (Cardoso et al., 2004) that compares workflow management systems (WFMSs) and enterprise resource planning (ERP) systems is referred in the theoretical analyses and is further developed. Furthermore, when comparing the two types of systems in theoretical perspective, two criteria are taken into account for selecting the dimensions used for comparisons:

1. The first criterion is that the chosen dimensions should enable the comparisons to provide deep insight into the characteristics and functioning of the systems, thus the readers can have a thorough comprehension of what the systems are and their differences.
2. The second criterion is that the chosen dimensions should evaluate the systems from a practical application perspective, thus the readers can obtain the knowledge of how the systems work and how they differentiate from each other.

To meet the first criterion, two dimensions are selected for comparing the systems: the first is the "domain scope" which describes the suitability of the systems for a specific type of application or organization; the second is the "technological scope" which describes the technological capabilities of the systems. Two other dimensions to satisfy the second criterion are selected for analyzing the systems: the first is the "lifecycle scope" which describes the procedures of applying the systems; the second dimension is the "knowledge translation" which describes the methodology of translating paper-based clinical guideline into executable models in the systems. Some of the dimensions are further explained below:

- The "domain scope" dimension includes two aspects: application goal (what is the system used for in general); application independency (whether the system is domain-independence or domain-specific).
- The "technological scope" dimension discusses only one aspect: type of support (what object does the system support, process or data).
- The "knowledge translation" dimension includes three aspects: guideline representation (how does the system represent information obtained from the clinical guideline); modeling methods (how does the system build models); process definition (how the process is being defined, is it explicit or implicit).

3.3. Practical implementations

The practical implementation is to further verify the theoretical analyses in the previous section, as well as to re-examine the hypotheses proposed. The dimensions used to measure the outcome of practical implementation of guidelines cases in both clinical decision support system and workflow management systems are described in the last part of this section.

Based on literature studies, basic knowledge of both systems' characteristics and functionalities can be acquired. Given the research objective, a complex clinical guideline case will be used for the comparison of practical implementation. Considering that Gaston as a guideline-based clinical decision support system is already used in several departments in Catherina hospital Eindhoven, in order to obtain more real-life data from the practical use of the system, clinical guidelines that are supported by Gaston will be considered first. After discussing with medical experts from the hospital, one specific clinical guideline case will be acquired and used for implementation. Reasonable complexity will be considered as next criteria in selecting the guidelines.

Before selecting which workflow management systems are for practical implementation, a characterization of clinical processes and their match to support by the certain workflow management system will be introduced, combined with the theoretical comparisons, decision will be made on the selection of suitable workflow management systems for practical implementation and for further analyses.

After the practical implementation of the clinical guideline cases in workflow management systems and Gaston, evaluations will be made based on observations of the implementation. The evaluations are conducted in two perspectives: one is from the author's experience with using the systems, the other one is from the interview with experts and their comments on the systems. The analyses will be made based on three dimensions: functionalities of the system, advantages and disadvantages, and user perspective. The first two dimensions are in accord with the goal of the project; the third dimension is to evaluate the practical use of the system in real-life from user’s perspective.

4. Background Knowledge

In this chapter, the background knowledge of three types of flexible workflow management systems, Gaston knowledge-based clinical decision support system and categorizations of clinical processes is introduced. The first section depicts a certain category of workflow management systems, flexible workflow management systems, which demonstrates adequate support to flexible processes, e.g. clinical processes. Although there are several workflow management systems, their flexibility types are different. Thus this section will look into and compare the different flexibility type and the application fields of three flexible workflow management systems: YAWL, FLOWer and Declare, which are the most popular flexible WFMSs; the second section focuses on Gaston as a knowledge-based clinical decision support system, which is a quite successful CDSS that is applied in Catherina Hospital Eindhoven, its technical infrastructure and functionality are described; the third section gives a general categorization of clinical processes, and from the categorization given by previous research, the selected clinical process for the practical implementation can be categorized, also the most suitable workflow management system to support the selected clinical process can be decided and is tested in the implementation of the system(s).

4.1. Workflow management systems and flexibility types

In this section, the characteristics of flexible workflow management systems (YAWL, FLOWer and Declare) will be described in detail. YAWL, FlOWer (BPMone control part) and Declare are the flexible workflow management systems that aim at supporting processes with different types of flexibility. YAWL was chosen because it is a powerful open-source system supplying most of the workflow patterns. FLOWer is considered to be the most successful commercial system providing flexibility. Declare is an academic system providing new and powerful ways of supporting “extreme” flexibility that can cope with most exceptional situations.

4.1.1. Workflow-patterns-based WFMS (YAWL as representative)

YAWL (Yet Another Workflow Language) is an open source workflow management system, which is based on the well-known workflow patterns. YAWL supports the modeling, analysis and enactment of flexible processes through the use of worklets and which can be seen as a kind of configurable process fragment. YAWL is built on top of Petri nets, but in order to overcome the limitations of Petri nets, it has been extended with constructs to facilitate patterns involving multiple instances, advanced synchronization and cancellation patterns. YAWL is inspired by Petri nets but is not just a macro package built on top of high-level Petri nets: it’s a completely new language with independent semantics (Van der Aalst & Ter Hofstede, 2005). Moreover, YAWL allows for hierarchical decomposition and handles arbitrarily complex data.

Figure 7 shows the modeling elements of YAWL. At the syntactic level, YAWL extends the class of workflow nets with multiple instances, composite tasks, OR-joins, removal of tokens, and directly connected transitions.

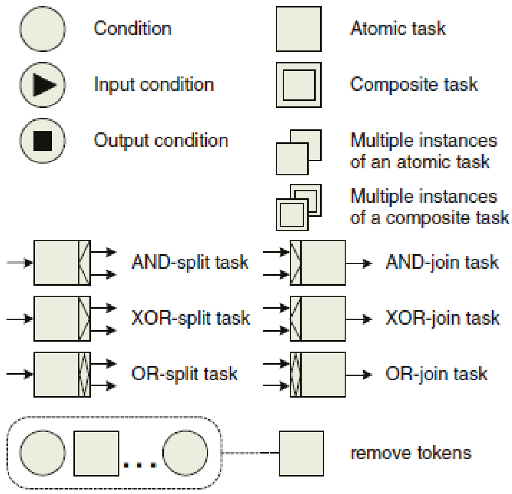


Figure 7: Symbols used in YAWL

The architecture of YAWL workflow management system is depicted in Figure 8 (Van der Aalst, Aldred, Dumas, & Ter Hofstede, 2004), it consists of five main components: YAWL designer, YAWL manager, YAWL repository, YAWL engine and YAWL services.

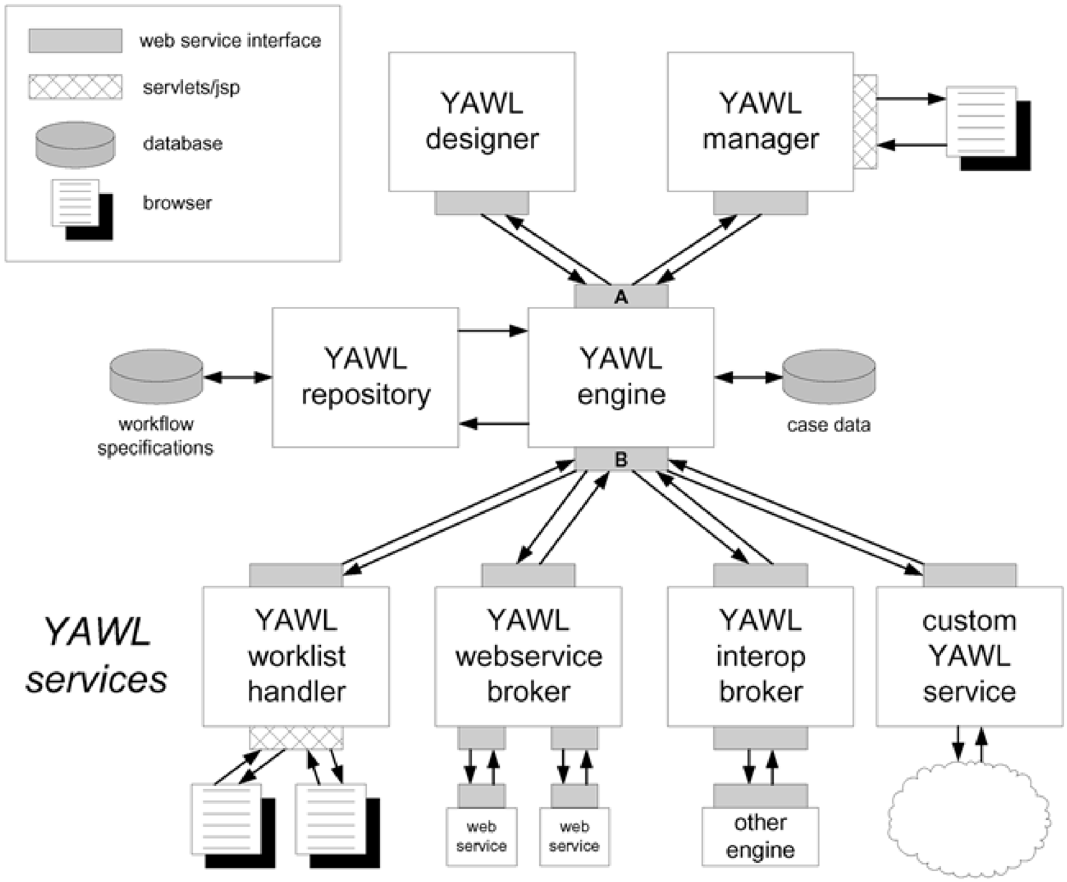


Figure 8: Architecture of YAWL system

YAWL designer is for designing workflow specifications, which will be deployed into the YAWL engine to perform all necessary verifications and task registrations before being stored by

YAWL repository. Workflow specifications can be instantiated through the YAWL engine, leading to workflow instances (also called cases). The engine handles the execution of these cases, i.e. based on the state of a case and its specification, the engine determines which events it should offer to the environment, in this case, the so-called YAWL services.

YAWL services include: YAWL worklist handler, YAWL web services broker, YAWL interoperability broker and custom YAWL services. The YAWL worklist handler is the component used to assign work to users of the system, through which the users can accept work items and signal their completion. The YAWL web services broker is the bridge between the engine and other web services, it acts as a mediator between the YAWL engine and external web services that may be invoked by the engine to delegate tasks (e.g. delegating a "payment" task to an online payment service). The YAWL interoperability broker is a service designed to interconnect different workflow engines, for example, to subcontract a task to another system where the task corresponds to a whole process. A custom service connects the engine with an entity in the environment of the system. For example, a custom YAWL service could offer communication with mobile phones, printers, assembly robots, etc.

4.1.2. Case Handling WFMS (FLOWer as representative)

Case Handling refers to the ability of a workflow system to allocate the work items within a given case to the same resource at the time that the case is commenced. It is a specific approach to work distribution that is based on the premise that all work items in a given case are so closely related that they should all be undertaken by the same resource. Case handling is a new paradigm for supporting flexible and knowledge-intensive business processes. Unlike workflow management, which uses predefined process control structures to determine what should be done during a workflow process, case handling focuses on what can be done to achieve a business goal. There are four core features of case handling (Van der Aalst, Weske, & Grünbauer, 2005):

- Avoid context tunneling by providing all information available (i.e., present the case as a whole rather than showing just bits and pieces);
- Decide which activities are enabled on the basis of the information available rather than the activities already executed;
- Separate work distribution from authorization and allow for additional types of roles, not just the execute role;
- Allow workers to view and add/modify data before or after the corresponding activities have been executed (e.g., information can be registered the moment it becomes available).

Different from workflow management supported by contemporary workflow technology, case handling focus on the whole case. Its primary driver to determine which activities are enabled is the state of the case (i.e., the case data). The logistical state of a case (i.e., which activities are enabled) is derived from the data objects present, therefore data and process cannot be separated. Moreover, it allows for a separation of authorization and distribution and is possible to distinguish various types of roles. (Van der Aalst et al., 2005) has summarized those main differences in table 3 as below:

	Workflow management	Case handling
Focus	Work-item	Whole case
Primary driver	Control flow	Case data
Separation of case data and process control	Yes	No
Separation of authorization and distribution	No	Yes
Types of roles associated with tasks	Execute	Execute, Skip, Redo

Table 3: Differences between workflow management and case handling

There are several tools that can suit the case handling paradigm. In this project, BPMone from Pallas Athena is selected for study. BPMone from Pallas Athena is divided into two parts: BPMone Design and BPMone Control. BPMone Design is similar to Pallas Athena's prior product Protos, whereas BPMone Control is comparable to the prior product FLOWer. BPMone is used as a workflow management system, but the BPMone Control part specifically supports the case handling paradigm. FLOWer consists of a number of components: FLOWer Studio, FLOWer Case Guide, FLOWer Configuration Management (CFM), FLOWer Integration Facility, and FLOWer Management Information and Case History Logging.

BPMone Design is a modeling environment for designing the processes. Once you have modeled your processes in BPMone Design, BPMone offers a simple way of seeing whether the processes are actually implemented in the way they were intended, by converting the model to a workflow model that can be accessed in BPMone Control. BPMone Control can access to the workflow model created in BPMone Design and to keep everything organized: a different "worklist" (a sort of to-do list) can be created for each user, the level of access for each user can be defined, the specific forms the user will see can be adjusted and placed into the desired sequence, skip/redo options can be linked to the specific user who can apply them, and much more.

4.1.3. Declarative WFMS (Declare as representative)

Declare is a prototype of a workflow management system that based on constraints. Instead of explicitly defining the ordering of activities in models, the models of Declare rely on constraints to implicitly determine the possible ordering of activities, in other words, any order that does not violate constraints is allowed. Figure 9 specifies the difference between traditional imperative approach and constraint-based declarative approach. An imperative model usually specifies exactly how to execute the process while a declarative model focuses on a set of constraints. In this way, the declarative model implicitly defines the control flow as all possibilities that do not violate any of the given constraints, as shown in Figure 9 (Pesic, Schonenberg, & Van der Aalst, 2007).

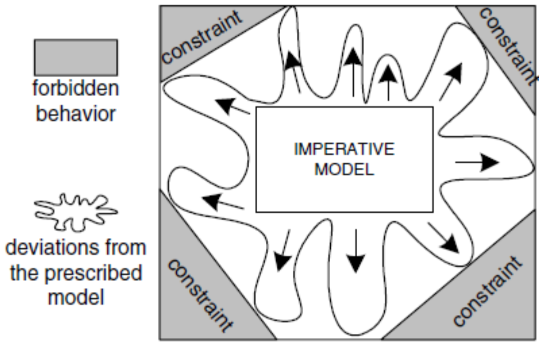


Figure 9: Imperative vs. declarative approach

To realize the declarative approach, Declare uses a customizable set of arbitrary constructs called constraint templates. Templates are defined on the system level in the Designer component. Each template has a unique name, a graphical representation, and a formal specification of its semantics in terms of Linear Temporal Logic (LTL).

Figure 10 (Van der Aalst et al., 2009) depicts the architecture of Declare system. The core of the system consists of the following basic components: Designer, Framework and Worklist. The Designer component is used for creating the so-called constraint templates, to design concrete process models, and to verify these models. The Framework enacts instances of process models. Moreover, it also conducts ad-hoc changes of running instances. While the Framework centrally manages the execution of all instances, each user uses his/her Worklist component to access active instances. Also, a user can execute activities in active instances in his/her Worklist.

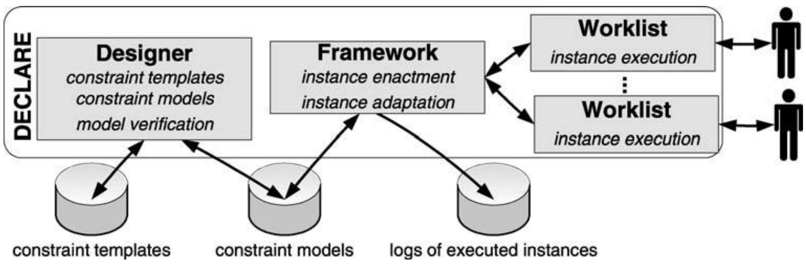


Figure 10: The architecture of Declare

4.1.4. Flexibility types

In the literature study, several types of workflows are distinguished. Among the various types of workflows, declarative workflow and case handling workflow are defined as new paradigms for supporting flexibility. The case handling paradigm focuses mainly on the case itself rather than emphasizing the routing between atomic activities and being strongly process-oriented. It aims at what can be done to achieve a business goal. FLOWer as the control part of BPMone is the workflow management system based on the case handling paradigm. The declarative workflow is the paradigm that makes it possible to balance between support and flexibility. It uses constraint-based models that implicitly specify the execution procedure by means of constrains: any execution that does not violate constraints is possible. Declare is thus grounded on the declarative workflow paradigm. In process-aware information systems, various perspectives can be distinguished. The control-flow perspective captures aspects related to control-flow dependencies between various tasks (e.g. parallelism, choice, synchronization, etc.). The data perspective deals with the passing of information, scoping of variables, etc., while the resource perspective deals with resource to task allocation, delegation, etc. YAWL is the workflow management system that built based on the patterns of control-flow perspective (44 patterns in total) and is considered as a quite flexible workflow management system since it can support most of the patterns.

For the control flow perspective, there are several process flexibility types, which provide means for business processes to respond to changes in their operating environment: Flexibility by Design, Flexibility by Deviation, Flexibility by Underspecification and Flexibility by Change. The various flexibility types and their use scope (the situations and domains to which the flexibility type applies) are summarized below (Schonenberg, Mans, Russell, Mulyar, & Aalst, 2008):

- **Flexibility by Design:**
  Definition: The ability to incorporate alternative execution paths within a process model at design time allowing the selection of the most appropriate execution path to be made at runtime for each process instance.
  Scope: Any process which may have more than one distinct execution trace.
- **Flexibility by Deviation:**
  Definition: The ability for a process instance to deviate at runtime from the execution path prescribed by the original process without altering its process model.
  Scope: Particularly suited to the specification of process models, which are intended to guide possible sequences of execution rather than restrict the options that are available.
- **Flexibility by Underspecification:**
  Definition: The ability to execute an incomplete process model at run-time.
  Scope: Mostly suitable for processes where it is clearly known in advance that the process model will have to be adjusted at specific points, although the exact content at this point is not yet known.
- **Flexibility by Change:**
  Definition: The ability to modify a process model at run-time such that one or all the currently executing process instances are migrated to a new process model.
  Scope: Flexibility by change allows processes to adapt to changes, which may be introduced both at instance level, and type or model level, that are identified in the operating environment.

In the research work of (Mans, Van der Aalst, Russell, Bakker, et al., 2009), researchers have evaluated the degree of support for the four flexibility types in YAWL, FLOWer and Declare. The evaluation results are illustrated in table 4 by indicating whether the workflow system support (+) or does not support (-) the respective flexibility type. None of the evaluated systems provide the full range of flexibility alternatives. Flexibility by design is supported by all offerings. YAWL excels in flexibility by underspecification, FLOWer in flexibility by deviation and Declare excels in deviation and change.

Flexibility by	YAWL	FLOWer	Declare
Design	+	+	+
Deviation	-	+	+
Underspecification	+	-	-
Change	+	-	+

Table 4: Flexibility types of WFMSs

4.2. Knowledge-based CDSS (Gaston as representative)

Gaston actually can be interpreted as both guideline representation model and clinical decision support system, which belong to the stages of interpreting and implementing clinical guidelines. As a guideline representation model, it facilitates the development of clinical guideline application tasks. It covers all stages in the guideline development process, ranging from knowledge representation and acquisition, guideline verification to guideline execution by a decision support system. Gaston consists of a suite of tools that support several in the guideline development (Figure 11) (De Clercq, Blom, Hasman, & Korsten, 2000). It combines domain ontology and method ontologies to translate a paper-based clinical guideline into executable models. Different from common clinical decision support systems which only apply primitives and Problem Solving Method (PSM), Gaston combines both of them in translating clinical knowledge. PSMs pre-define the global control structure, in a Knowledge Acquisition Tool the author only has to specify the knowledge component, while in the primitive-based approach the control structure has to be explicitly stated. Primitives are usually used in a sharable guideline representation to represent steps such as actions, decisions and plans. PSMs represent generic strategies to solve stereotypical tasks, independent from the system's application domain. One advantage of the PSM-based approach is the separation of domain-specific knowledge and domain-independent methods, which increases the reusability and share ability.

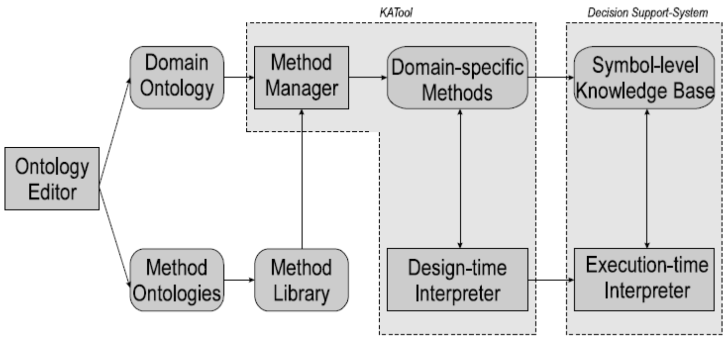


Figure 11: Process view of the GASTON framework

There are four stages involved in the process:

- Stage 1: Defining domain and method ontologies. Develop, derive or reuse application-specific domain and method ontologies.
- Stage 2: Developing method libraries (A method library is a collection of Problem Solving Methods and primitives created in the second stage that can be used in the third stage

for the definition of guidelines to solve certain tasks). Develop or reuse libraries of Problem Solving Methods.

- Stage 3: Authoring guidelines. Develop guidelines in terms of Problem Solving Methods and primitives through a Knowledge Acquisition-Tool.
- Stage 4: Executing Guidelines. Automatically translate these guidelines into a more efficient symbol-level representation, which can be read in and processed by an execution-time interpreter.

To have a clear overview of the architecture of Gaston, figure 12 displays its components as below. It consists of a collection of modular components for the application of guidelines with design-time components, execution-time components and interface components. (De Clercq et al., 2000)

The aim of design-time components is to facilitate the guideline authoring process based on guideline representation models along with execution-time components for building decision support systems that incorporate these guidelines. The interface components allow a knowledge engineer to define the specifications for communication between the execution-time components and other systems, such as Electronic Patient Record (EPR) systems. Within interface components, the Event part defines the specification of event that triggers the execution of guidelines while the Action part provides the interface for communications between the user and the decision support system. It defines the means how the decision support is delivered to the user (e.g. alerts, reminders, or even an interactive dialogue).

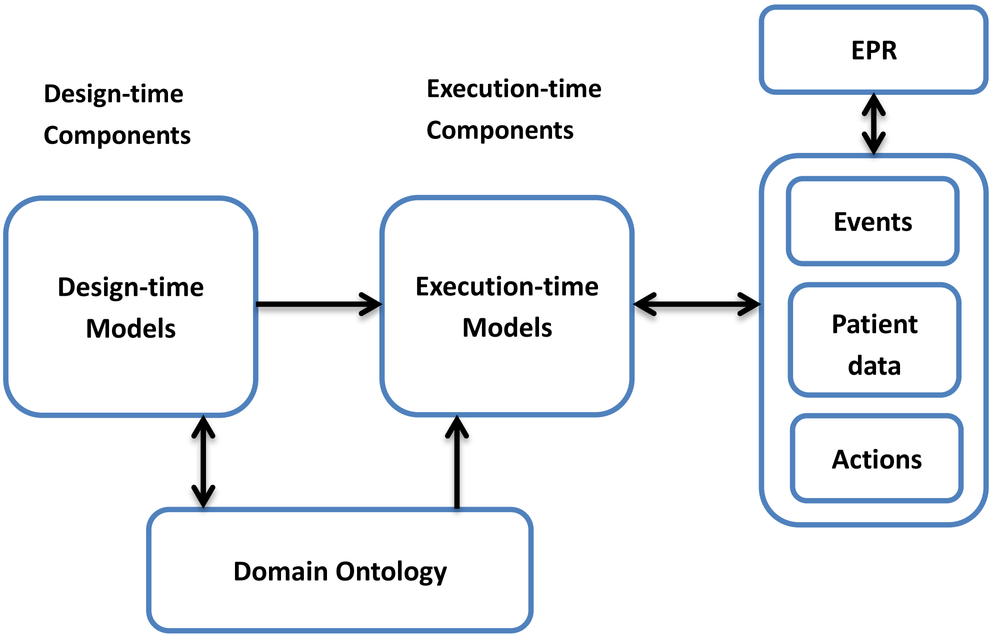


Figure 12: Gaston Architecture

In stage 3, the KA-Tool is used to facilitate the guideline authoring process. An overview of the components in KA-Tool is shown in Figure 13. It consists of a collection of modular components, in which the core is a design-time interpreter. The design-time interpreter acts as a bridge between user and various knowledge managers such as a Domain Manager, a Guideline Manager and a Method Manager. The design-time interpreter combines the information from all available managers and creates a user interface that enables guideline authors to develop guidelines in terms of PSMs and primitives.

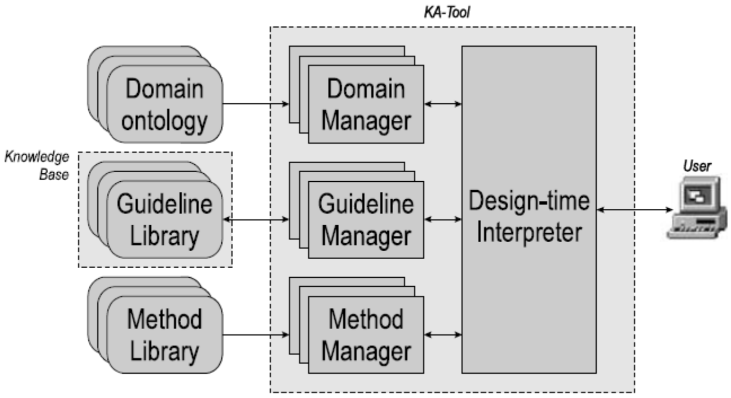


Figure 13: System overview of the components in the KA-Tool

In stage 4, the decision support system plays a major role in supporting the execution of guidelines. Figure 14 depicts the components in Gaston decision support system. It consists of a collection of modular components with an execution-time scheduler and four plugins. The Execution-time Scheduler functions by communicating with the plugins through a standard interface that does not differentiate between different types of plugins. The Procedure Manager Plugin communicates between the compiled structure and Execution-time Scheduler and informs the Execution-time Scheduler which guidelines are to be executed and in what order. Datasource Manager makes use of standard communication protocols to exchange clinical data with data sources such as EPRs that contain clinical data. The Event Manager Plugin defines the specification of event that triggers the execution of guidelines. The Action Manager Plugin establishes the means of communications between the user and the decision support system. It defines the means how the decision support is delivered to the user.

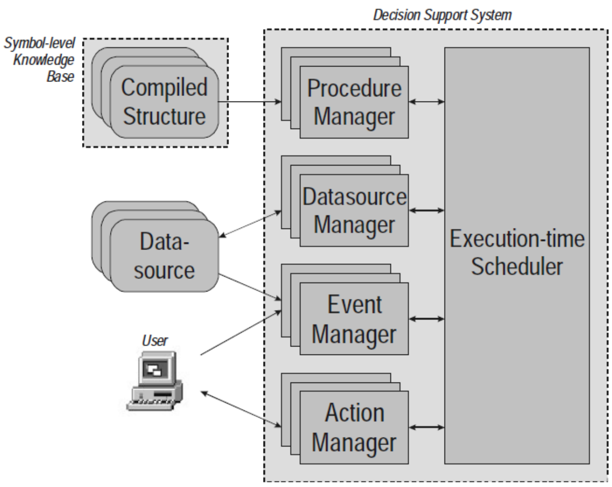


Figure 14: System overview of the components in Gaston Decision Support System

4.3. Clinical process support

There exist many healthcare processes with totally different characteristics for which other requirements with regard to flexibility will exist. Figure 15 illustrates healthcare processes based on care types, complication probability, diagnosis and complexity of care.

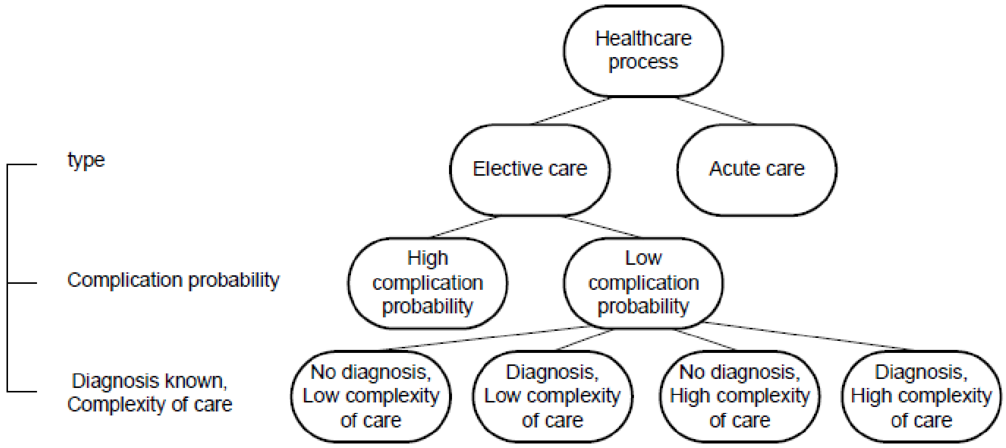


Figure 15: Categorization of clinical processes (Mans, Van der Aalst, Russell, & Bakker, 2009)

Acute care deals with critically ill patients in which patient conditions change rapidly. Consequently, acute care cannot be planned in advance and needs to be done in an ad-hoc fashion. To this end, flexibility by change is required for such ad-hoc process as the model is not fixed and can be changed into another completely specified model, thereby guaranteeing model correctness (Mans, Van der Aalst, Russell, & Bakker, 2009). However, elective care can be planned in advance but still classifications can be identified. Elective care with high complication probability will cause a high impact on the process as it requires the process to be changed dramatically in some parts. The kind of flexibility needed for this situation is provided by flexibility by change (Mans, Van der Aalst, Russell, & Bakker, 2009). On the contrary, elective care with low complication probability requires less change to the processes. For the elective care with low complication probability, the complexity of care can be distinguished. Complexity of care indicates the extent of care, which is delivered to a patient, which can be either high or low. If the complexity of care to be delivered is low only a few departments are involved in diagnosing and treating the patient. More or less, a standard process can be followed for diagnosis and treatment. Same for the diagnosis process, if a diagnosis is known, the process to treat the patient will also be clear. In order to depict these processes in a complete model, a high amount of flexibility by design is needed. For some parts of the process, decision may depend on intermediate results of diagnostic tests or patient's reaction to treatment offered, etc. In this case, process should be partly modeled as a deferred choice, whose ultimate realization can be deferred until runtime. Mans et al. have researched on the flexibility types of workflow management systems and the six categorizations of health cares. Table 5 below summarizes which flexibility approach is considered important and more relevant to which kind of healthcare process.

Categorizations of care /Flexibility by	Design	Deviation	Underspecification	Change
Acute care				**X**
Elective care				
High complication probability				**X**
Low complication Probability				
Low complexity, no diagnosis known	**X**	**X**		
Low complexity, diagnosis known	**X**	**X**		
High complexity, no diagnosis known			**X**	
High complexity, diagnosis known			**X**	

Table 5: Flexibility types match to categorizations of healthcare process

Combining with the classification of workflow management systems based on flexibility types, systems that can best support for which kind of healthcare process can be reasoned. As FLOWer supports both flexibility by design and flexibility by deviation, it is the best candidate for healthcare process with low complexity. For healthcare process with high complexity, flexibility by underspecification is needed, thus YAWL will be the most suitable system. Finally, for both acute care processes and processes with a high complication probability, flexibility by change is required, Declare can provide "extreme" flexibility to the process, thus it is the best candidate in this situation.

4.3.1. Clinical guideline of anti-coagulation

In this section, the anti-coagulation clinical guideline for CTC patients before and after the operation is introduced. The anti-coagulation guideline aims at providing a guideline for patient CTC in terms of safe discontinuation of anticoagulation preoperatively and in terms of optimum adjustment of the anticoagulant within the safe therapeutic range. The clinical process of anti-coagulation falls partly into acute care according to the medical expert that works with the anti-coagulation guideline and sometimes the guideline should not be fully depended on due to exceptional situations.

The starting point of the processes is the patients pre-admission. After that, patients need to go through pre-operation screening, planning operation, medication review, admission, medication use, INR test and therapy in order to make sure the patient is ready for operation before they can be operated. After the operation, patients need to use medication and have INR test to ensure stable health condition. If the INR level is within the target range, patients can be discharged from the hospital.

In Figure 16, the simplified version of anti-coagulation clinical guideline flow chart is depicted and the full version of the flowchart cannot be presented due to privacy. The simplified version flow chart is originally made by a nursing specialist from the cardiothoracic surgery department of Catherina hospital and is revised based on discussions between the nursing specialist and author of the thesis. The total processes include more than 20 tasks; some of the tasks are repetitive and will be performed for several times (e.g. INR test). Some of the tasks are optional and will be performed depending on the state of patients (e.g. based on the INR level, different medicine will be used for treatment), most of the tasks are sequential, which follows the procedural work in the clinical processes. Since there are some inevitable exceptional situations existing in the processes, and they are not described in the standard clinical guideline process, experience and knowledge are required by clinicians for coping with those situations. In the models, the exceptional situations are considered as much as possible, due to different modeling mechanisms and system nature, processes will not be fully covered, but the performance of the systems will be analyzed and evaluated. Based on the interview with the expert from the cardiology department in Catherina hospital, the anti-coagulation process falls into the categorization of an elective clinical process with low complication probability but high complexity, but the preoperative preparation and intensive care process sometimes will be the acute process. Due to these characteristics, the best candidate of the system to support the whole process would be YAWL and Declare.

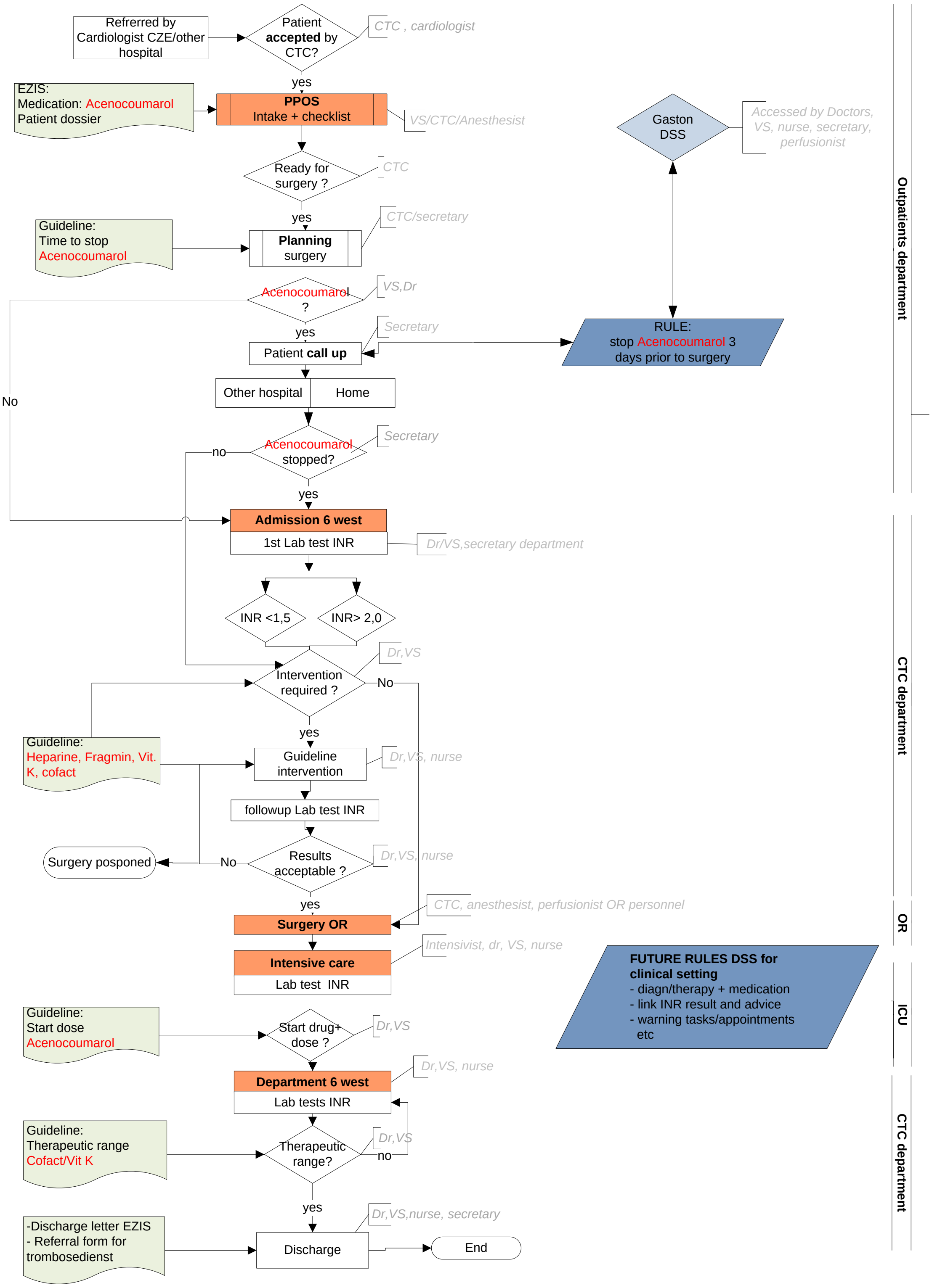


Figure 16: Simplified version of anti-coagulation clinical guideline

To perform those tasks, four departments are involved and take the responsibility, namely Outpatient Department, CTC Department, ICU department and Operation department, where nine roles of people participate in this process, including Anesthetist, Cardiologist, Intensivist, Perfusionist, operation personnel, Doctor, Nurse, Secretary and Nursing Specialist (NS). The organizational context is illustrated by Figure 17.

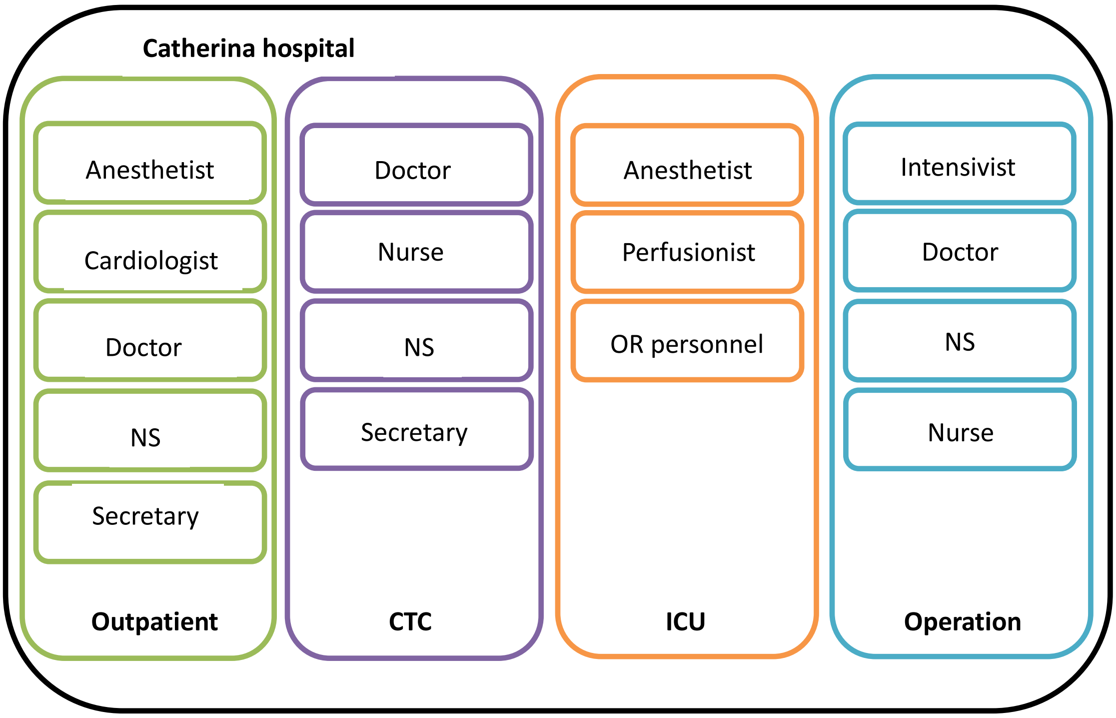


Figure 17: Organizational structure of anti-coagulation guideline in Catherina hospital Eindhoven

5. Theoretical Analyses of Flexible WFMS and Gaston

In this chapter, the analyses are made based on the literatures of workflow management systems and clinical decision support systems.

The theoretical comparisons of the two systems mainly discusses the comparisons between Gaston, which is a guideline-based clinical decision support system used in Catherina hospital Eindhoven, and the three flexible workflow management systems. The comparisons will be conducted based on four perspectives: first and second perspectives concentrate on domain scope and technological scope of the systems; third and fourth perspectives focus on the lifecycle of using the systems and their knowledge translation. The four perspectives are chosen due to two reasons: on the one hand, a deeper understanding in terms of the applications, infrastructures and characteristics of the systems can be acquired; on the other hand, theoretical knowledge of using the two types of system can be learned.

5.1. Theoretical comparisons of WFMSs and Gaston

After the overview of differences between WFMSs and CDSSs in chapter 2, and comparisons among three types of WFMSs (YAWL, FLOWer and Declare), now it comes to more detailed conceptual comparisons between the three types of WFMSs and one specific CDSS (Gaston). Firstly, analyses will be conducted based on two scopes: domain scope and technological scope. Each part, or dimension, contrasts the two species of systems from a specific point of view. Afterwards, descriptions will be made in terms of the use of both systems from a life-cycle point of view and the knowledge translation of both the systems.

5.1.1. Domain Scope

The first dimension analyzes both systems according to their domain scope. The domain scope defines the suitability of a system for a specific type of application or organization. This characterization is important since organizations have different needs and characteristics.

	YAWL	**FLOWer**	**Declare**	**Gaston**
Application goal	Coordination and automation of tasks	Coordination and automation of tasks	Coordination and automation of tasks	Provide clinicians with warnings of clinical tasks
Application independency	Domain independence	Domain independence	Domain independence	Domain independence

Table 6: Domain scope comparisons

Workflow systems are composed of applications and tools that allow for the definition and management of various activities associated with workflows. A workflow normally comprises a number of tasks, dependencies among tasks, routing rules and participants. A task can involve human performance or can be executed automatically by IT applications. A workflow system reads, automates, processes, and manages workflows by coordinating the sharing and routing of information (Cardoso et al., 2004). Thus the process efficiency is improved by the automation of work items. Gaston as a guideline representation model aims to facilitate the development and implementation of computer interpretable guidelines. Clinical guidelines or clinical practice guidelines are "systematically developed statements to assist practitioner decisions about appropriate health care for specific clinical circumstances" (Grossman & Field, 1990). The main issue tackled by Gaston is the representations and share of various types of guidelines using a formal and unambiguous framework. By executing the computer interpretable guidelines, Gaston is able to provide warnings or messages to the users, thereby the accuracy and safety of clinical decisions can be improved.

Workflow systems have been installed and deployed successfully in a wide spectrum of organizations. Workflow systems enable developers to separate the flows among a system's components (application and data) from the workflow process model, making these applications flow-independent. The flow-independence concept makes workflow technology suitable to a large number of domains. Therefore, the generic tool constituted in WFMSs can be used to integrate different types of data and applications in a broad spectrum of contexts, including healthcare, military, administrative processes, etc. While some workflow management systems are generic, others are more oriented to particular domains. Specifically, Declare provides full support for loosely-structured processes. YAWL and FLOWer are more generic WFMSs that can support either standardized or flexible processes. The case-handling paradigm in FLOWer enables human actors to have an overview of the complete process, including completed and future activities, and it limits the context tunneling effect to a large extent.

The framework of Gaston as a clinical decision support system consists of a newly developed guideline representation formalism to represent guidelines of various complexity and granularity and different application domains and guideline authoring and execution environment. In Gaston infrastructure components, the guidelines are modeled in an editor before contained in a reposition, which is separated from data and other applications. The architecture of the guideline execution engine is system independent as well as application-independent so that the guideline engine can be used in multiple clinical domains. Compared to workflow management systems, which depend on process model, Gaston relies on clinical guidelines and is able to support wide spectrum of domains. However, since Gaston is currently being used by physicians in hospital for clinical decision-making, whether it can be used for business organizational decision support is not known.

5.1.2. Technological Scope

The second dimension, technological scope, is used to compare workflow management systems and Gaston. This dimension characterizes the systems based on their technological capabilities.

	YAWL	**FLOWer**	**Declare**	**Gaston**
Type of support	Workflow support, Process-centric	Workflow support, data-centric	Workflow support, Process-centric	Decision support, Patient/data-centric

Table 7: Technological scope comparisons

Workflow management systems and Gaston have been developed with distinct sets of technological capabilities. The goals of WFMSs are the coordination of human tasks, and the integration of independent, heterogeneous and often distributed information systems (HAD), data and applications (overview of WFM). In particular, workflow technology allows for the integration of previously separate communication, information and data flows into a working process at any time.

Instead, Gaston as a clinical decision support system does not focus on coordinating and integrating roles, tasks and applications, but it focuses on a specific case or patient case. It will collect data of a certain patient and make some calculations based on computer interpretable guidelines before giving advice to the clinicians. So compared to most workflow management systems, which are process-centric, it is a data-centric type of system. FLOWer as a case-handling system is an exception (Reijers, Rigter, & van der Aalst, 2003), it is data-centric due to the fact that the state of a case within a case-handling system consists of the complete collection of information on the case, including workflow control data and application data.

5.1.3. Lifecycle scope

The third dimension is the lifecycle scope. Workflow management systems are useful tools in the field of business process management (BPM). There are several phases in the BPM life-cycle (Figure 18), workflow management systems start from design phase of the whole phases.

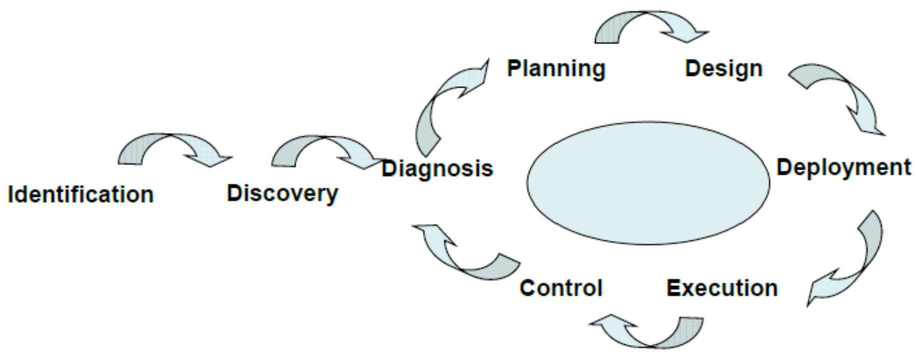


Figure 18: BPM lifecycle

Before start modeling the processes, key activities should be first found. This step is included in the Identification and Discovery phase. After selecting the process, process models consist of key activities are built in the editor of WFMSs. In the design phase, Declare is quite different from the other two WFMSs. Besides modeling the whole processes, it also supports scenario modeling where flexibility is mostly required. For each activity or task in the process, the flow definition and data definition need to be specified so that data can be transferred between the activities or tasks, also roles for performing the specific task should be defined. After validating the process model using validation tools, the process model can be executed by the predefined roles. Some workflow management systems include tools for analyzing the process. For example, YAWL has a small tool for the specification analysis; some workflow management systems connect with external analysis tool, and Declare usually connects PROM for process mining and other analyses. Through the analysis tool, problems exist in the process can be discovered, for instance, the bottleneck activities, the activity with least or most resource utilization, etc. Thus the process can be changed for improvement.

For Gaston as a guideline-based clinical decision support system, the procedures are quite different (Figure 19). The starting point is to translate a paper-based guideline to a computer interpretable guideline (CIG) or clinical rules by using Gaston guideline representation formalism in Gaston authoring environment, normally the modeled clinical rules are not like process models that are complex and consist of multiple procedural tasks but are more close to "decision trees" that are precise and simple. And then the clinical rules will be verified and tested in a simulation environment before they are executed by physicians. After the check of clinical rules, decisions will be made whether they need refinement or not.

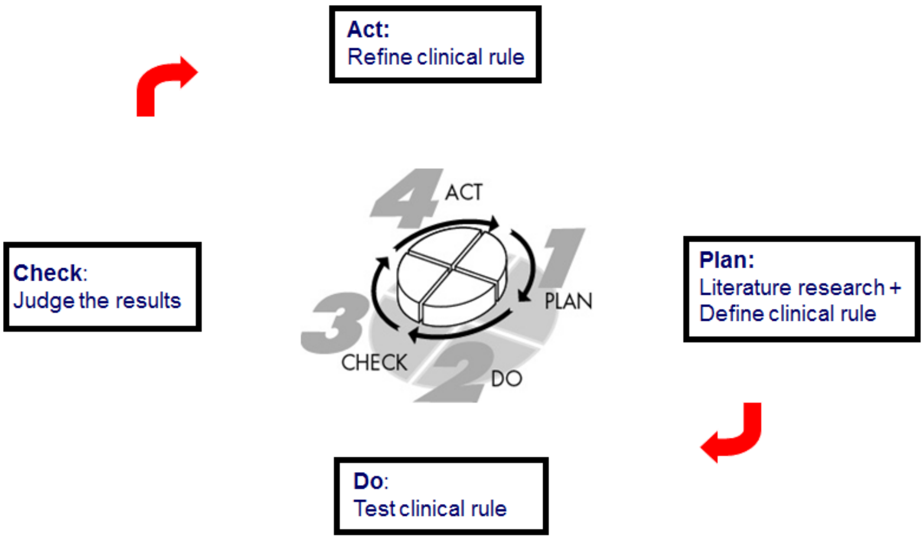


Figure 19: Development of clinical rules in Gaston

5.1.4. Knowledge translation

The fourth dimension is the "knowledge translation". In this section, guideline representations as well as the modeling languages in the three workflow management systems and Gaston will be described. In other words, how the system represents information obtained from the clinical guideline will be explained in this section.

	YAWL	FLOWer	Declare	Gaston
Guideline representation	Process model	Process model	Process model	Decision tree flow chart
Process definitions	Explicit process definition	Explicit and implicit process definition	Implicit process definition	Explicit and implicit definition
Modeling language	Workflow pattern	Case handling	Declarative workflow	Gaston representation

		model

Table 8: Knowledge translation comparisons

In terms of the guideline representations in the three types of workflow management systems, YAWL models the guidelines based on petri-nets but extend with features that facilitate more workflow patterns involving multiple instances, advanced synchronization patterns, and cancellation patterns. The process definition in YAWL is mostly explicit. It needs to specify all possible orderings of the related tasks. But under some circumstances, such as time-out scenarios, YAWL do not need to add explicit routing constructs, simply offer tasks, which can handle these things like providing timed tasks. FLOWer is similar to the WFMS approach in that various activities in the process definition are distinguished, but only a preferred or normal control flow between these activities is modeled. The process modeling of FLOWer is implicit, since the designer has to specify what is not permitted. It offers more flexibility by adding special actions that users can perform while working with imperative models. By default, at run time an actor is able to execute, redo, or bypass these activities, possibly diverging from the normal flow. Declare is developed as a constraint-based system and uses a declarative language grounded in temporal logic for the development and execution of process models. In this way, the declarative model implicitly defines the control-flow as all possibilities that do not violate any of the given constraints.

The guideline representation formalism developed in the GASTON framework combines the concepts of primitives, which are used to construct the guidelines control structure explicitly, and problem solving methods (PSMs) which usually describe knowledge on an abstract level and are used to model guidelines that perform stereotypical tasks. The clinical guidelines are developed in terms of PSMs and primitives through a Knowledge Acquisition Tool (KA-Tool), and will be automatically translated into a more efficient symbol-level representation.

In order to cater for the flexibilities in healthcare processes, workflows types such as declarative workflow are further developed by researchers in order to meet the uncertain requirements of healthcare institutions. As a consequence, WFMSs that can support flexible workflow model are developed, for instance, YAWL, BPM|One and Declare workflow management systems are dealing with adaptive, case-handling and declarative workflows respectively. The three systems will be selected as the objects to be further investigated in the project and based on the result of theoretical study, one or two of the workflow management systems will be selected in practical comparison.

After theoretical analysis of the three types of workflow management systems, two of them will be selected for practical comparison with clinical guideline cases to further validate the theoretical comparisons.

6. System Implementations of Flexible WFMS and Gaston

In this section, a clinical guideline of anti-coagulation will be implemented both in YAWL and Declare workflow management system and Gaston clinical decision support system. After this, detailed descriptions of the implementations will be given, including data definition and model design construct, followed by an evaluation of the execution in both systems. The evaluation contains two aspects: design evaluation and execution evaluation with respect to the designing and executing process respectively. The implementation is presented to medical experts during the interviews. All the models have been validated by the experts and their comments of the systems are considered in the evaluations. In terms of three flexible WFMSs that are analyzed in the theoretical part, two of them are further used in the practical implementation. FLOWer is excluded due to a practical reason that the server is not accessible in the project period of modeling.

6.1. YAWL implementation

The data type definition, design constructs and execution scenarios of the anti-coagulation clinical guideline in YAWL will be explained in this section.

In Figure 20, the main process of the guideline is given. It is also the topmost page of the processes based on the clinical guideline of anti-coagulation in Figure 16. The topmost processes include three sub-processes: “Therapy”, “INR_Test” which is contained in sub-process “Therapy” and “Intensive_Care”. The workflows of all the sub-processes in anti-coagulation clinical guideline are presented in Appendix A.

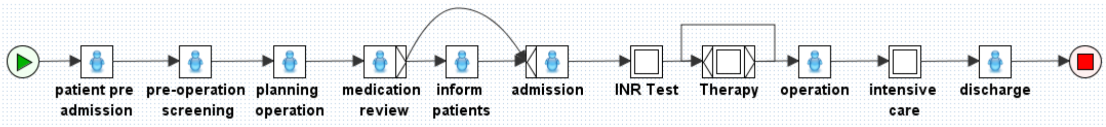


Figure 20: Top-level YAWL process

6.1.1. Data definition

While the initial focus of YAWL is on control perspective, it is also fully supporting the data perspective. YAWL completely relies on EXL-based standards like XPath and XQuery, thus it’s possible to define data elements and use them for conditional routing, for the creation of multiple instances and exchanging information with the environment, etc.

Appendix 2 shows the declaration of variables for the anti-coagulation clinical guideline. Part of the definitions is depicted in the picture below. In all of the data types of the YAWL model, three of them belong to simple data type, including “patientNotification”, “patientCondition” and “postponeOK” and five of them are complex type, namely “patient”, “subscription”, “operation”, “screening”, “INR_Level” and “medicationChecklist”. The complex data type contains two or more simple data types or refers to another complex data type. Here the most important data element is the “PatientData” which is a complex data type. This data element contains patient personal information, medical record, test results and operation time, etc. and it exists in almost every task and flow definition.

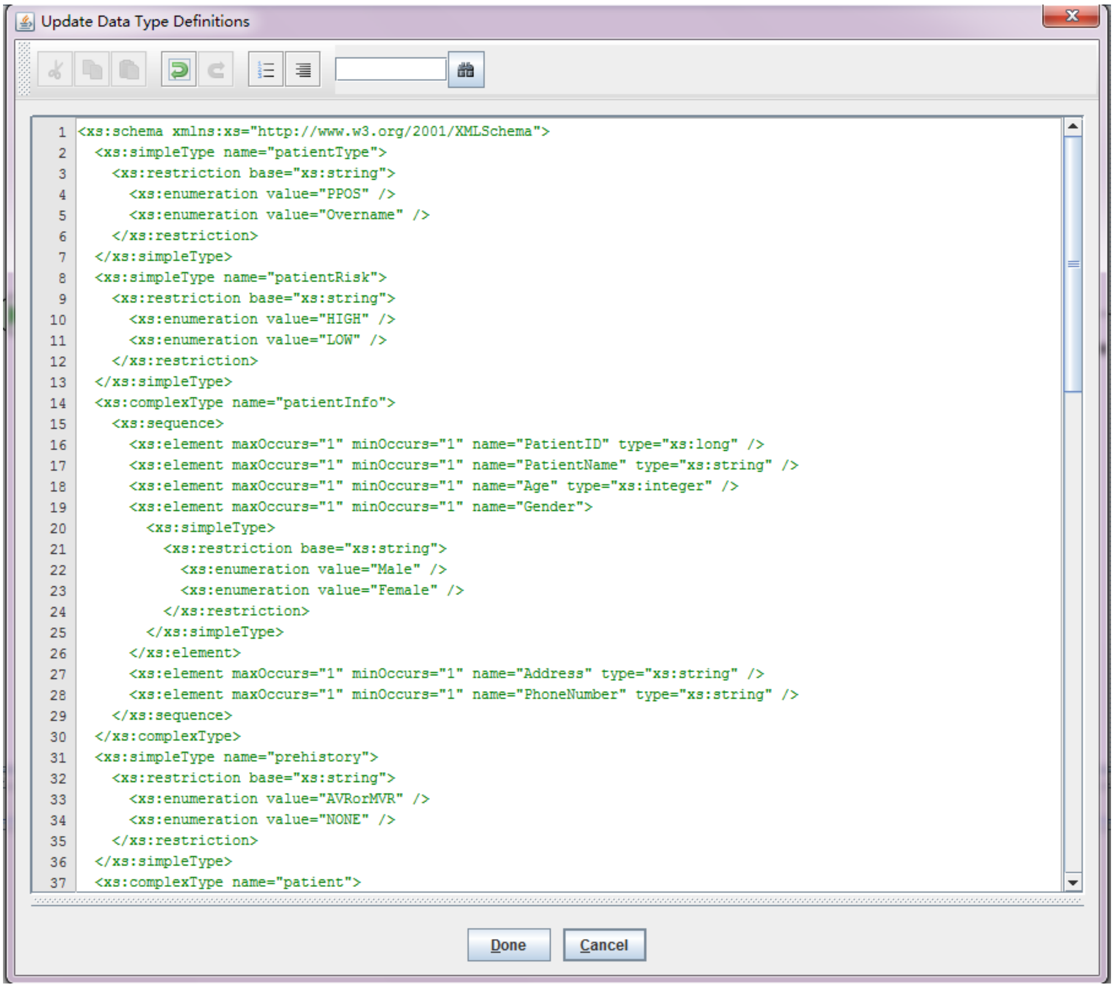


Figure 21: Data definition in YAWL

6.1.2. Design constructs

In YAWL system, the processes require explicit modeling due to the fact that it is best used for supporting highly structured and standardized processes. In terms of the anti-coagulation clinical guideline, relevant activities are modeled in a sequential order, some exceptional situations can be modeled by adding extra tasks and arcs, for example, in the “INR test” sub process, if the tested INR level of the patient is too high, say above 3.0 (normal range is between 1.5 to 2.0), then the patient needs to retest after sometime, thus the condition “more than 3.0” and timed task “retest after one day” are modeled after task “test INR1” (Figure 22).

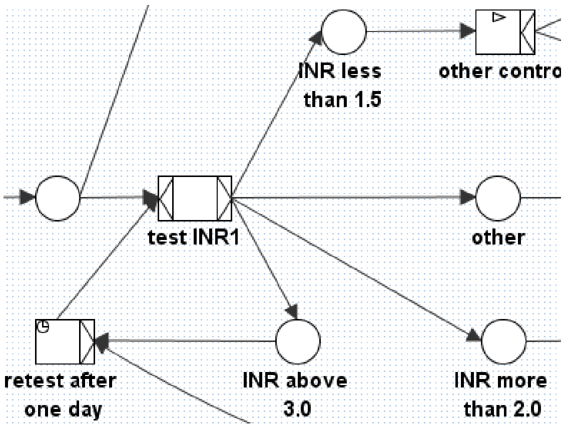


Figure 22: Decision task and out-going routes

But there are still some exceptional situations that cannot be modeled in YAWL. For example, if the patient is under emergency call, the tasks of registration, pre-operation screening, medication review and inform patients can be skipped, the patient will be admitted and planned an operation as soon as possible. Meanwhile, tasks after admission may not follow the sequential order as modeled in YAWL.

At the registration step, patient information is input manually into YAWL system, thus the later tasks can retrieve information for automatic routing. In general, patients need to take pre-operation screening to check whether health condition is ok for operation, then the electronic medical review (EMR) should be examined by doctor. If the patient takes some specific medicines, e.g. Acenocumarol, he should be informed in advance to stop the medicines several days before operation. A few days before operation, patients are admitted into the hospital and take the INR test (Figure 23). After entering the result of INR level, task "test INR1" then retrieves INR result and is automatically routed to the corresponding treatment measure. The task of "test INR1" is modeled as a XOR-Split task so that one of the different routing of INR results can be reached. Simultaneous treatment measures (start Heparine and control APTT) are modeled after a AND-Split task "other control".

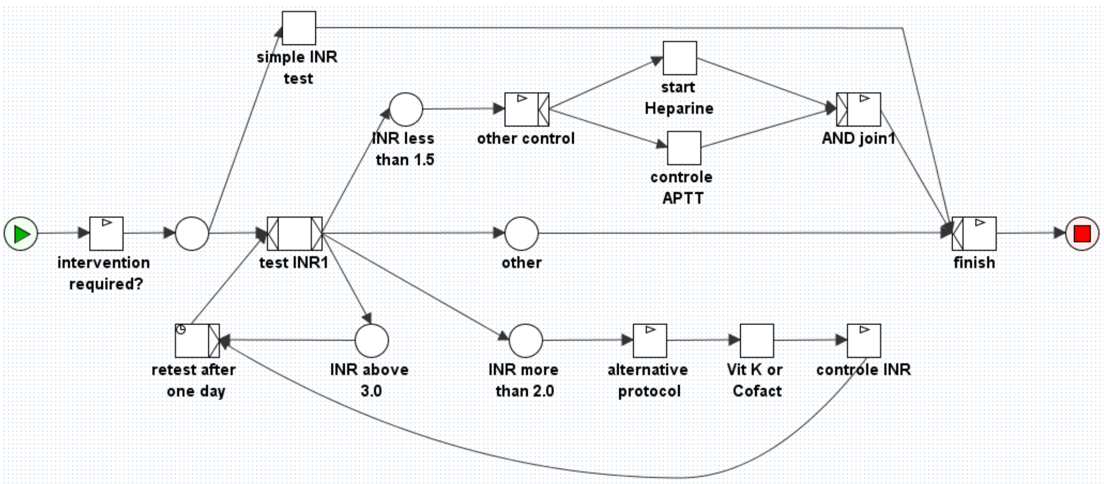


Figure 23: INR test processes

After the INR test, some criteria will be judged by the nurses before taking other treatment measures, for instance, whether the patients are with prehistory aortic valve replacement (AVR) or mitral valve replacement (MVR), whether the patients use coumarin derivatives (e.g. Acenocumarol) and whether the patients are with high risk. Based on second INR test result, doctor will decide whether the patient is suitable for operation. If not, the operation will be postponed. For modeling the postpone operation scenario, a cancellation task is applied. if the task of postpone operation is executed, tasks "stop Heparine before operation" and task "ready for operation" will be cancelled.

Task "result acceptable" use a OR-Split task.

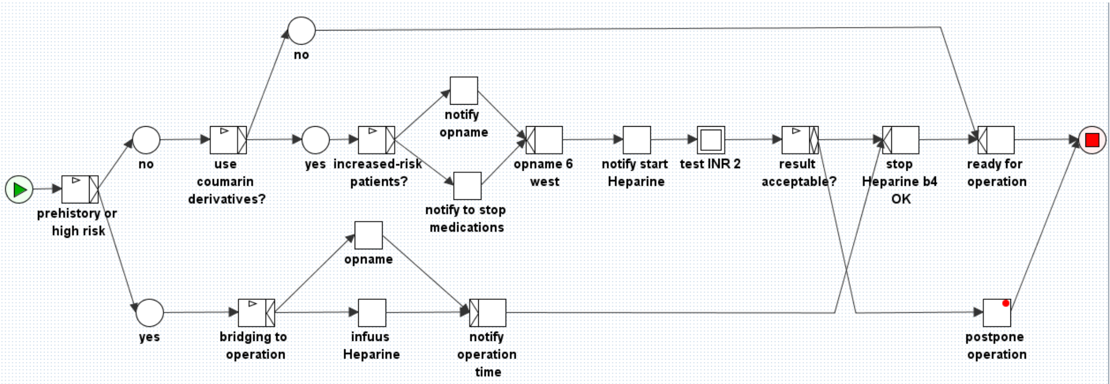


Figure 24: Therapy process

In the process of intensive care, which is after patient taking operation, patients still need to be checked the INR level before they can leave the hospital. Here the scenario is that the INR level of the patient has to be repeatedly checked in order to maintain it in the normal range. Thus a timed task "time counter" is added after check INR. A timed task will be executed after the predefined time period is passed. In this case, if the INR level is not ok, doctor can wait until the time passed and the system will directly lead to the step of test INR. If the doctor thinks that patient is ready for dischargement, he should execute the following task "notify trombosedienst" before time counter reaching the time period.

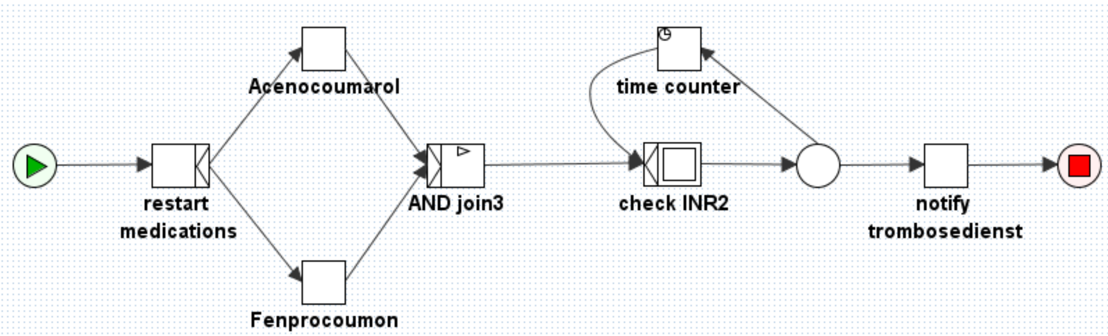


Figure 25: Intensive care processes

6.1.3. Analyses of YAWL implementation

YAWL as a workflow-pattern based workflow management system has played the role of coordinating works and people in the clinical process of anti-coagulation. Compared to the visio flowchart of the anti-coagulation guideline (Figure 16), the process models in YAWL defined the sequence of tasks more explicitly and strictly. There are 45 activities are defined in the model, with specifications of 15 data definitions and more than 40 arcs. After assigning roles to each non-automatic task, the workflow can be executed. During the execution, users can find their tasks in the work item panel of YAWL control center (Figure 26). By accept and start the offer, the user can work on the task allocated to him or her. After the user completes the task, task after this immediately starts and will appear on the work item panel of the next responsible people.

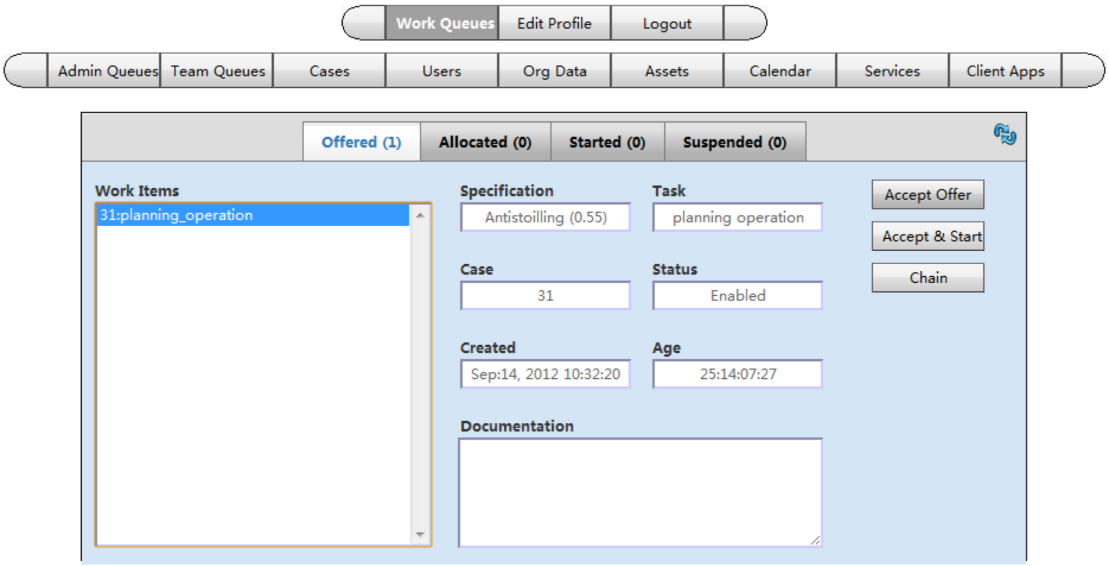


Figure 26: YAWL control center

The biggest advantage of YAWL workflow management system is that users can receive their task-to-do at the very first time on their own screen, thus the communication time spent on exchanging information of work progress can be greatly reduced compared the current situation that clinicians need to send paper documents to each other to confirm a clinical event. The second advantage of the system is credited to the timed task, which can function as a reminding of work to the users since the timed task will reappear on the work item panel after the pre-defined time period. For instance, in the anti-coagulation guideline, several INR tests are required to be done after surgery of patients, the clinicians thus need to check the INR level every day before they agree on the INR level of patients and discharge them from the hospital. By adding a timed task that will be enabled after one day, clinicians will receive “test INR” work item every day at a certain time as a reminder for them.

The disadvantages of YAWL are several. First of all, due to the nature of YAWL, it can only provide to users with the information of “which task to do next”, but cannot present users with the information of “what to do exactly”. Therefore, the users must have full knowledge of how to perform the tasks. This is not sufficient for supporting the work of clinicians. For example, in the guideline of anti-coagulation, before patients are admitted into the hospital, they should be notified of stoping one medicine three days before the surgery. However, in YAWL this message cannot be displayed during execution but only the task name "notify patients" could be shown. Secondly, although the data definition allows for multiple of data types as well as complex types, it doesn't allow the storage and retrieve of history data. For the same variable, every time a newer data is input by the user, it will replace the previous one. Thirdly, the timer function in YAWL can only count time afterwards but cannot count backwards, thereby for those clinical tasks that should be done, three days for example, before surgery, YAWL is not able to play the role of reminder. Fourthly, due to the fact that YAWL is best in support structured and repetitive processes, exceptional situations cannot be well modeled in YAWL. For example, in an emergency the patient will follow an acute care process which might skip several activities in the process; however, which activity to skip is dependent on the patient's condition, thus multiple of possibilities exist, for which YAWL cannot well support since otherwise a "spaghetti" net will occur. The last but not least functional disadvantage of YAWL is that it hasn't been integrated into the enterprise information system in the hospital, also it hasn't connect to the patient database and medicine database, thereby all the data should be input manually.

From the use perspective, it is not easy to start with modeling in YAWL system. Basic knowledge of Unified Modeling Language (UML) and XQuery is required for data definition and flow specification. The first step in modeling is to define the tasks and the routes thus to build the process models; the next step is to define all the data in a data definition panel (Figure 21), after this step, the tasks in the process can be connected to the defined data. For instance, as shown in figure 27 the first task "patient pre admission" is connected to two variables: PatientData and ScreeningT. The flow is also specified using XQuery as input or output parameters.

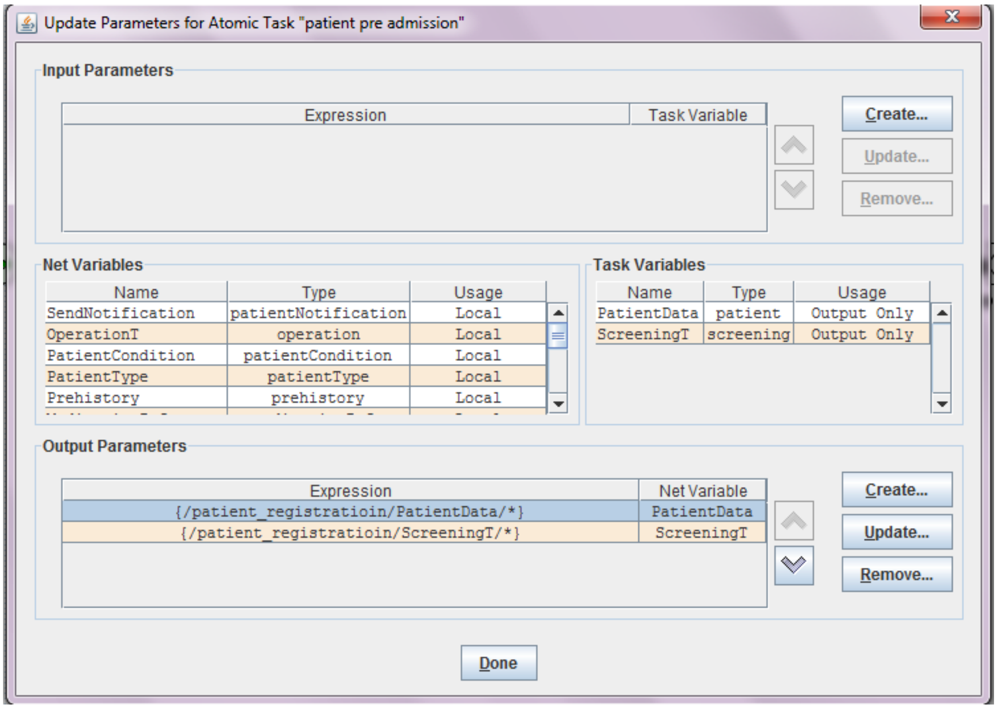


Figure 27: Task definition and flow specification interface

Every task except for automatic tasks needs specification of data and flow. All the data and flows are under strict specification, if any change is made to one of the variable, all the related tasks need to be re-specified in task variables and flows. This is very time consuming and troublesome. A good point of YAWL is that it has the validation tool which can help check missing data or flow definition. So the next step is to validate the process model, if it displays the information as Figure 28 below, it means that the model definition is correct. But whether it can be executed successfully is not confirmed. Other modifications might be made during simulations.

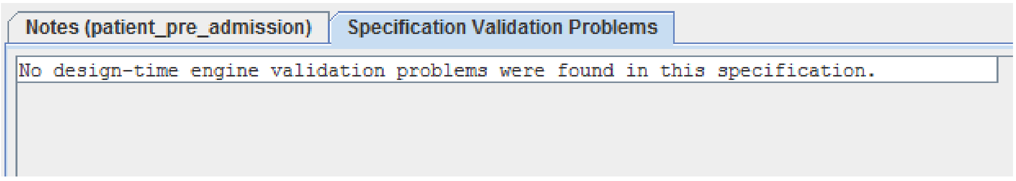


Figure 28: Validation result

To sum up, YAWL is a powerful workflow management system in coordinating tasks. In the anti-coagulation case, it helps allocate activities to responsible clinicians and is playing the role as reminding to-do tasks. However, since it is best in support strictly structured processes that have limited exceptional situations, the anti-coagulation processes are with acute care processes and thus are not fully supported by YAWL. To be maximally supported by YAWL, all the possible routes need to be pre-defined. Moreover, the definitions and specifications in YAWL are not simple work to do and are quite time-consuming. Therefore, if to apply YAWL in the hospital, YAWL expert is required to help medical expert in building the models.

6.2. Declare implementation

In this section, the data type definition, design constructs and execution scenarios of the anti-coagulation clinical guideline in Declare (version 2.2.0) are explained.

The process model of the anti-coagulation guideline in Declare is given in Figure 29, at first glance, it is much simpler than the process model in YAWL.

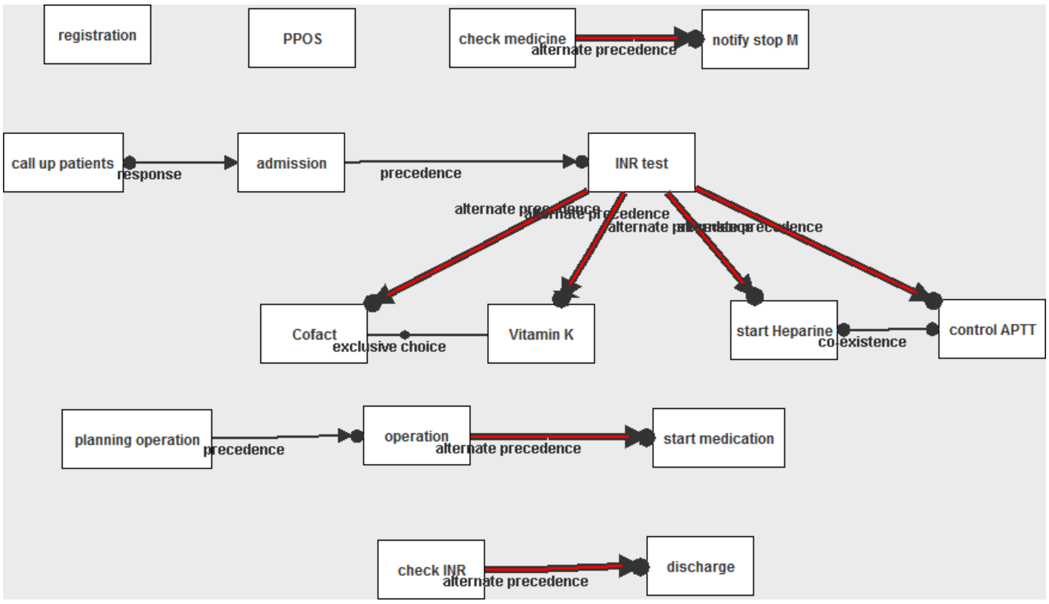


Figure 29: Process model in Declare

There are in total 17 activities in Declare compared to 45 activities in YAWL. Its simplicity is due to two reasons: the main reason is its constraint-based nature in modeling, besides the data definition is much simpler than YAWL. First, the data definition will be discussed. There are in total four data types in Declare: Integer, Double, Boolean and String. Because of limited data types, only few variables can be defined (Figure 30).

work | people | data

name	type	initial value
Patient ID	string	
Patient Name	string	
Use Acenocoumarol	boolean	
PConditionOK	boolean	
Operation Time	string	
INR result	double	0.0
Doctor Name	string	

Figure 30: Data definition in Declare

There are several constraint types in Declare, in the anti-coagulation process model, five constraints are applied, namely “response”, “precedence”, “alternate precedence”, “co-existence” and “choice”. The constraints are explained as below:

- **“Response”: Whenever activity "A" is executed, activity "B" has to be eventually executed afterwards.**

- **“Precedence”: "B" has to be preceded by "A". "B" can happen only after "A" had happened.**

- **"Alternate precedence": B cannot happen before A. After it happens, it cannot happen before the next A again.**

- **"Co-existence": If A happen (at least once) then B has to have (at least once) happpend before of has to happen after A. And vice versa.**

- **"Choice": Either A is executed at least once, or B is executed at least once.**

6.2.1. Design constructs

In the anti-coagulation clinical guideline, constraints exist on 14 tasks. Since patients can only be notified to stop medicine if their medication record has been checked, and the "check medicine" action won't definitely lead to "notify patient to stop medicine", thus precedence constraint is appropriate between the two tasks. Also "notify patient to stop medicine" cannot happen again before second time "check medicine" is happened, thus alternate precedence is best suited for this situation (Figure 31). The same situation occurs to tasks "INR test", "Cofact", "Vitamin K", "Start Heparine", "control APTT", "operation", "start medication", "check INR" and "discharge" as can be seen in the screen cut of Declare model.

Figure 31: Alternate precedence example

There are also two "precedence" constraints, and one of them is between "admission" and "INR test". The reason is that "INR test" can only happen after the patient is admitted into the hospital, but it is not necessary to happen again after second time of admission. The same explanation is for "planning operation" and "operation". There is only one "response" constraint which is between "call up patients" and "admission", since when patient is called up, they should be admitted to hospital soon, but there's also situation when patient is admitted without call-up, thus "response" constraint is best suited for this situation. Since in the anti-coagulation guideline, either "Cofact" or "Vitamin K" is used by the clinicians but not both, an "exclusive choice" constraint is used for modeling. And since "start Heparine" and "control APTT" should be used together, a "co-existence" constraint is used between the two tasks.

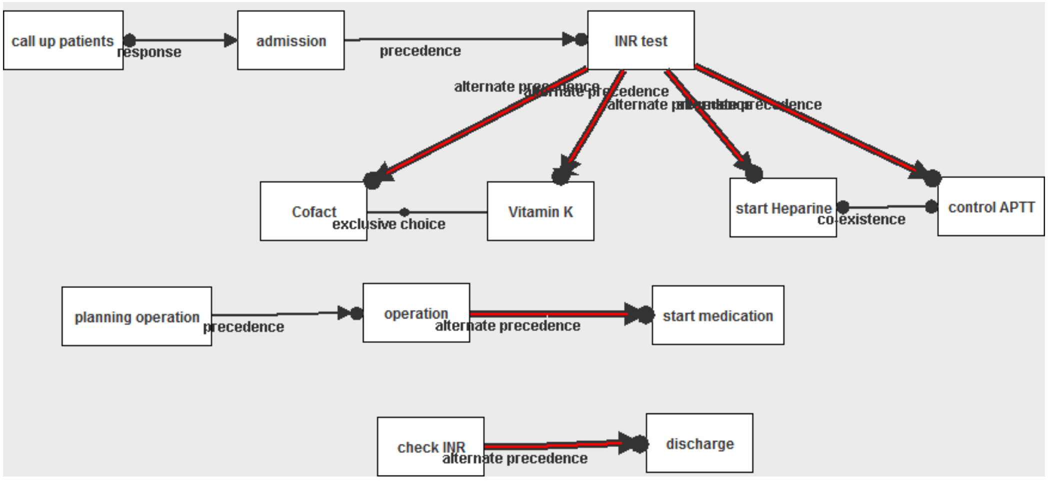


Figure 32: Constraints in Declare

6.2.2. Analyses of Declare implementation

Declare as a constraint-based workflow management system is best used for supporting loosely-structured processes. As can be seen from the model structure, the processes are built in an unstructured sequence, also for some of the tasks, there are even no arcs coming from and to them. But due to this loose structure, Declare is able to provide “extremely flexible” support to the processes.

The biggest advantage of Declare thus is the maximal flexibility it can provide. During execution, the starting task can be any task that is not constrained by other tasks. This feature is especially helpful to the acute processes in healthcare institution. Since when an emergency occurs, several tasks might be skipped and the sequence of executing the tasks might not be determined. In the anti-coagulation example, the preoperative preparation sometimes can be acute processes. In this case, the emergent patient should be admitted at the very first time and should be planned an operation time as soon as possible, or the patient is being planned an operation and admitted at the same. Declare is able to fully support this kind of situation, since the execution of task “admission” and task “planning operation” can be arbitrary. Task like “PPOS” which is a preoperative screening can be executed at any time.

The disadvantages of Declare are quite a few, it is because it’s still a prototype system which still needs much improvement. Firstly, the data types in Declare are very limited and simple. However, as in a complex environment as the hospital processes, they are not sufficient to meet the entire data requirement. Secondly, every task in the Declare model should be manually executed, not as in YAWL, some predefined flows can automatically check data and choose the route based on data level. Also the users should have full knowledge of the task to-be executed since the Declare is not able to provide messages to users with specific task content. Thirdly, as in the original anti-coagulation process, there are much more activities than in the process model of Declare, the reduction of activities would have an impact on the support of full processes since many details in the processes are neglected. Fourthly, the execution interface is the same as the modeling interface (Figure 33) and the only difference is that tasks are with “buttons”. Users execute the model by click on the “blue” buttons since the “grey” buttons are under constraints and cannot be executed. Although users can have a full picture of the processes during execution, mistakes might easily occur if the process is more complex. Compare with it, an interface with only the tasks that should be executed by the responsible user is displayed to him or her would be better.

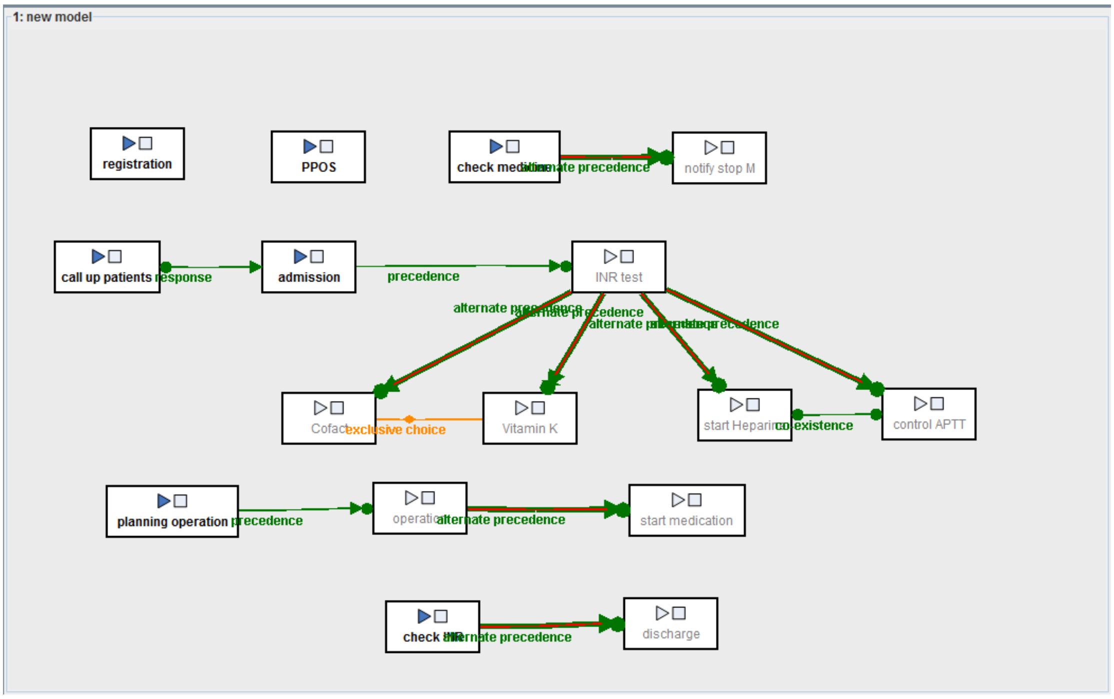


Figure 33: Execution of Declare model

From a use perspective, Declare is much easier than YAWL to work with. The main prior knowledge for users to work on Declare is to understand the different constraint types, then the users can build models in the designer. Since the data types are quite limited, data definition should not be complicated for users. Thus for the clinicians in the hospital, building models in Declare can be done by themselves.

To sum up, Declare as a constraint-based workflow management system has provided maximal flexibility to process support. The process model is simply structured and is easily to be built by users. However, due to fewer tasks and limited data definitions, the process model in Declare is rather abstract and thus is not able to support complex processes. The anti-coagulation clinical guideline contains moderate complex processes. Therefore, only part of the process, especially the acute care process, can be best supported by Declare. Other processes that are more structured cannot be sufficiently supported.

6.3. Gaston implementation

In this section, the data definition, design constructs and execution scenarios of the anti-coagulation clinical guideline in Gaston are explained.

Gaston has been used to develop, implement and evaluate guidelines and guideline-based decision support systems in several areas, including critical care, psychiatry, oncology, cardiology and chronic disease management. In the cardiology department of Catherina hospital Eindhoven, Gaston is used as a warning system to remind clinicians of critical issues. In terms of the anti-coagulation guideline, Gaston helps remind clinicians to inform patients of stop using coumarin derivatives (a kind of medicine for patients who have problems with cardiac valves) 2 or 3 days before the surgery. It is very critical information for the clinicians to conform to, since otherwise the surgery should be postponed.

6.3.1. Design constructs

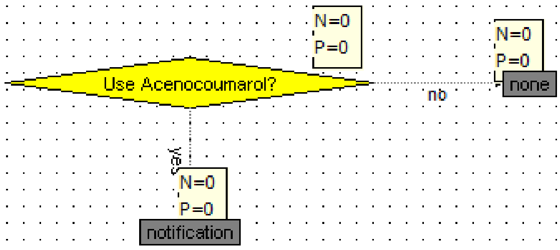
In this section, Gaston is used for re-modeling the clinical rules of anti-coagulation, and the implementation is tested by fake patient data. The design constructs in Gaston editor is described. Currently, Gaston has been used in the anti-coagulation guideline for checking three rules:

- Has Acenocoumarol been prescribed?
- Has Clopidogrel been prescribed?
- Has Fenprocoumon been prescribed?

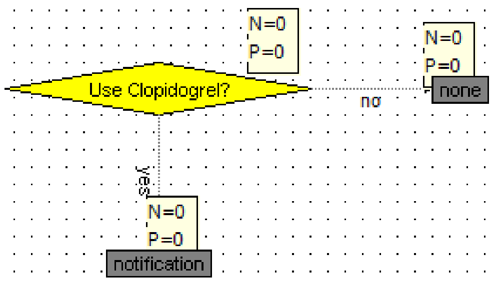
Acenocoumarol, Clopidogrel or Fenprocoumon are the medication names of coumarin derivatives for thinning the blood. When a patient is planned a date for operation, all three rules will be checked if the patient has been prescribed those anti-coagulation medication. If so, a warning is generated stating that the patient uses this medication, and that he should stop taking it 3 till 5 days before the operation takes place.

Gaston Editor is used for modeling clinical rules. The three rules used in the anti-coagulation guideline are built in the editor interface as below, they are in a uniform structure:

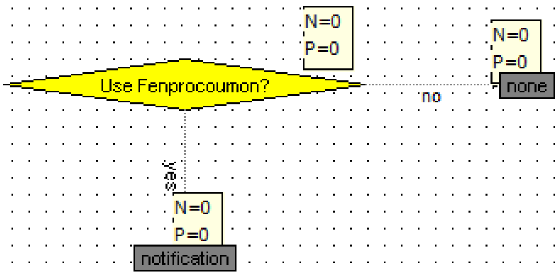
- Acenocoumarol guideline



- Clopidogrel guideline



- Fenprocoumon guideline



The yellow parallelogram stands for a decision step, where rules are checked for verity. In the first rule (use Acenocoumarol), as an example, the decision route is specified by adding drug name (Acenocoumarol) to its positive preferences (Figure 34). In this case, if the patient takes Acenocoumarol, the decision step will evaluate it as true statement and the corresponding recommendation that patient should stop using the medicine 2 days before operation is given. The recommendation is specified in the "notification" step.

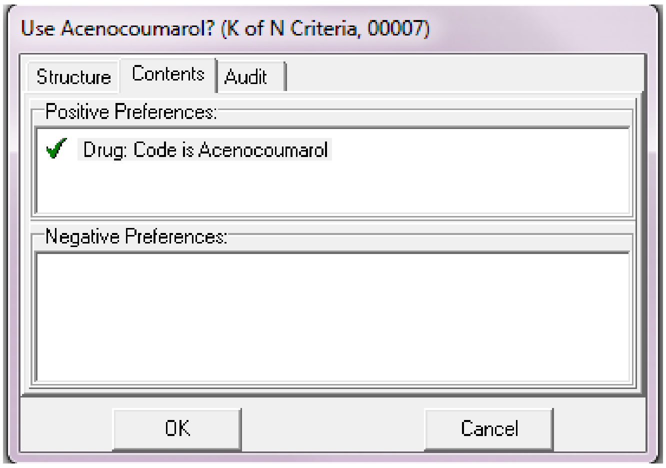


Figure 34: Decision step specification in Gaston

The recommendation for the patient who use those medications is specified in the "notification" step as in Figure 35 (Acenocoumarol as an example).

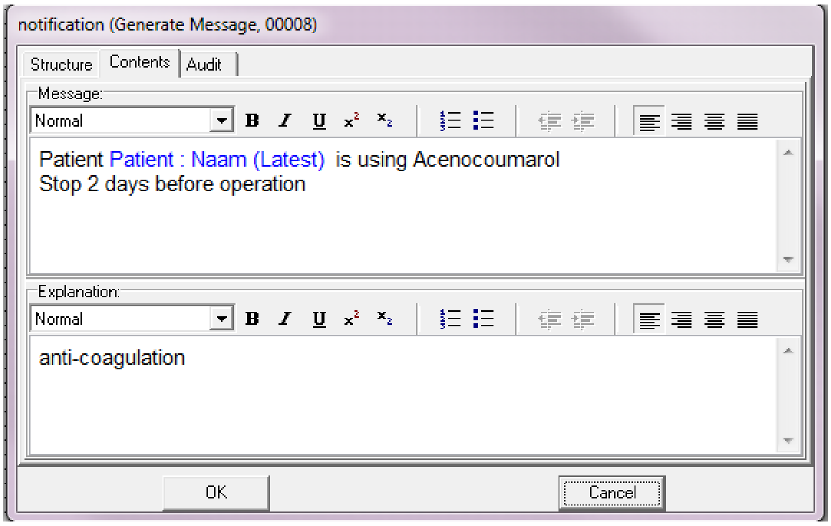


Figure 35: Recommendation step 1 specification in Gaston

For patients who are not using those medications, Gaston will not provide any information to the clinicians in practical use. But here in order to distinguish from the “notification” step where true statement is evaluated, a message indicating patient’s no use of medication is specified in “none” step as in Figure 36 (Acenocoumarol as an example).

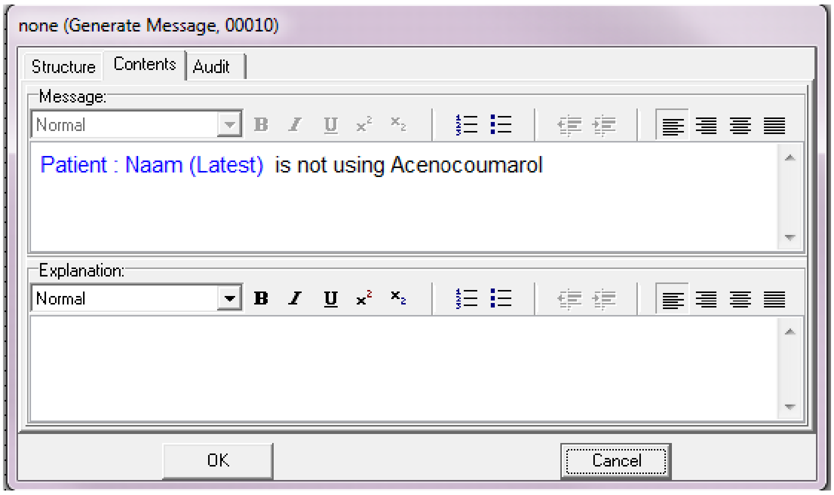


Figure 36: Recommendation step 2 specification in Gaston

The data input in Gaston can be either manual or automatically read from the database. The manual input of data can be achieved by adding an input step with specifications of patient information and medication information. But since Gaston is currently used with connection to the patient database in Catherina hospital, the implementation will also involve database and will be further described in next section.

6.3.2. Analyses of Gaston implementation

To implement the clinical rules built in Gaston editor, the first step is to enter artificial patient data into the database. Personal information of patients is filled in the Patient table, their medication information is input in the Drug table (Figure 37 & Figure 38).

CARRIE-PC\SQLEXP...an - dbo.Patient | CARRIE-PC\SQLEX...nShan - dbo.Drug

PatientID	PatientName	PatientMaidenN...	PatientFullName	roepnaam	voorletter	Gender
1234	Nan	NULL	Nan Shan	Shan	S	M
1235	Sam	NULL	Sam V	V	V	M
1236	Erik	NULL	Erik S	S	S	M
1237	Frank	NULL	Frank R	R	R	M
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 37: Patient information database

CARRIE-PC\SQLEXP...an - dbo.Patient | CARRIE-PC\SQLEX...nShan - dbo.Drug

PatientID	Code	ATCCode	GPKCode	Medicoms	Startdatumtijd	initieleStartdat...
1234	Acenocoumarol	NULL	NULL	NULL	NULL	NULL
1235	Clopidogrel	NULL	NULL	NULL	NULL	NULL
1236	Fenprocoumon	NULL	NULL	NULL	NULL	NULL
1237	Insulin	NULL	NULL	NULL	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 38: Drug information database

Here the information of four patients is entered as test data. Three of them take coumarin derivatives medication; each one takes one specific coumarin derivative medication. The fourth patient takes another medication. After starting Gaston Decision Support System (DSS) function, the warning information for each patient are displayed automatically as below (Figure 39-42).

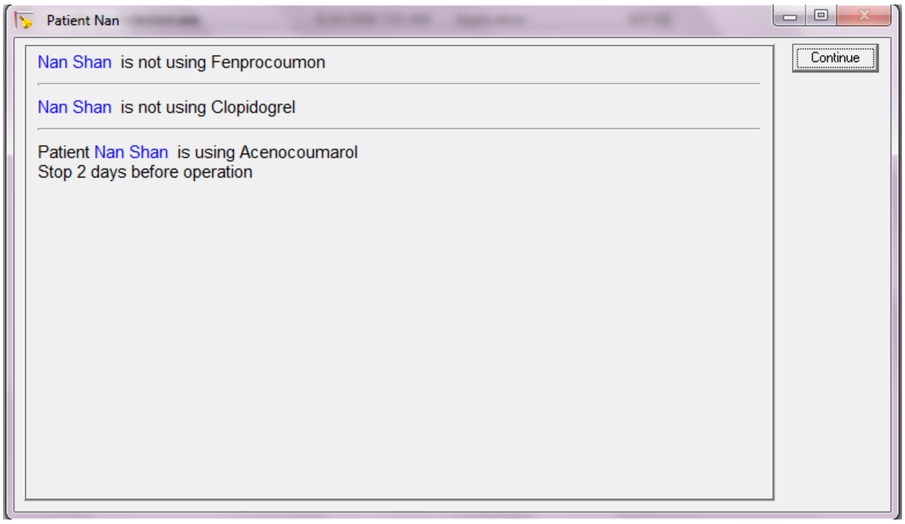


Figure 39: Pop-up warning for Patient Nan

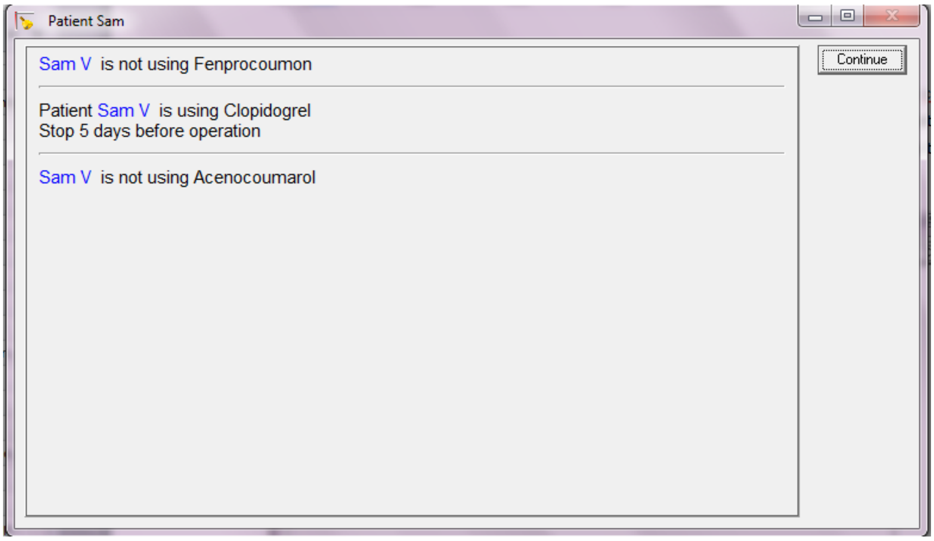


Figure 40: Pop-up warning for Patient Sam

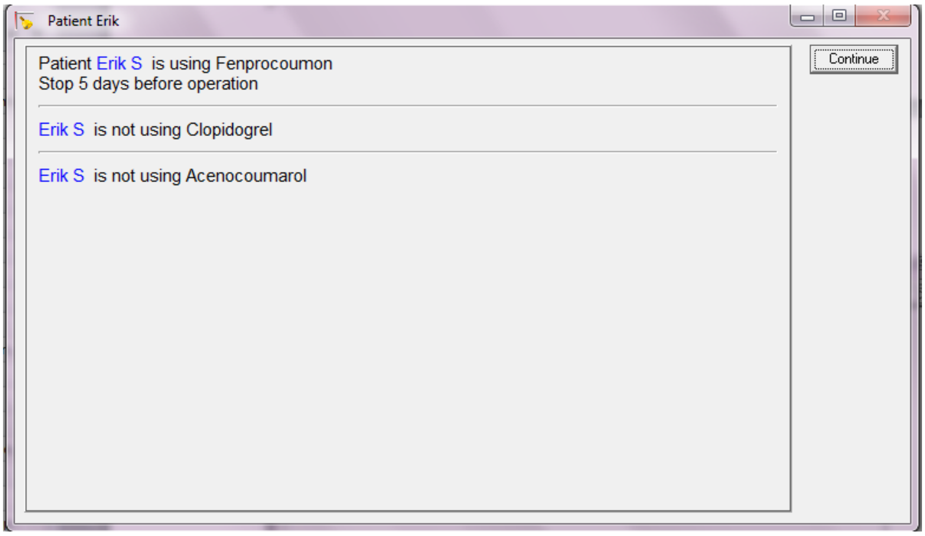


Figure 41: Pop-up warning for Patient Erik

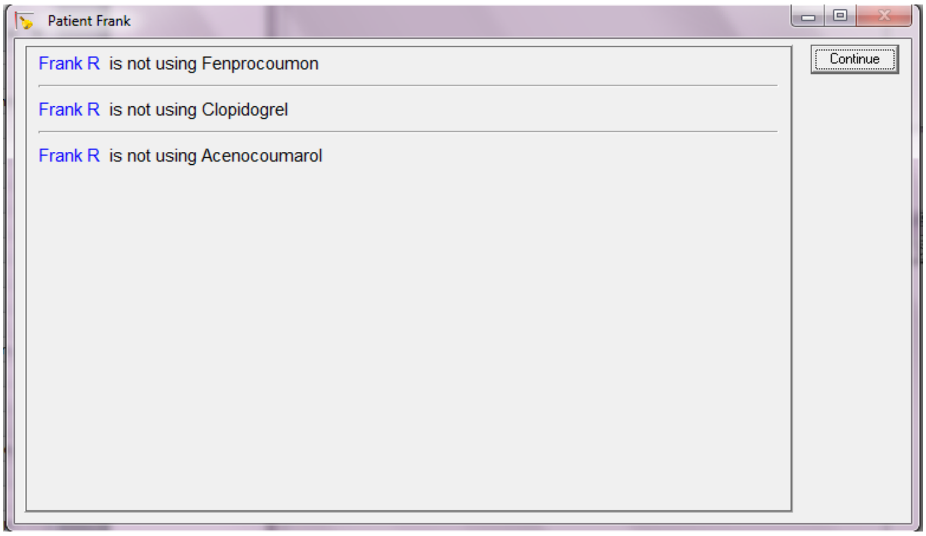


Figure 42: Pop-up warning for Patient Frank

As can be seen from the execution results, Gaston provide warnings to remind users of those patients who take coumarin derivatives. In Catherina hospital, Gaston is triggered manually by clinicians, usually under examination of one specific patient, thus Gaston will check the medication use of that patient in the electronic medical record (EMR) and then to display corresponding information.

In terms of the anti-coagulation guideline, Gaston as a guideline-based clinical decision support system has acted as a reminder of critical issues and an instructor of work to-do for clinicians. Its focus is not on the whole process but on the patient data, in other words, Gaston provides support to clinicians at a certain point of the process based on checking patient data. By providing warnings and messages to clinicians, Gaston aids to reduce clinical errors to a large extent. It's especially helpful for the inexperienced or occupied clinicians who might forget some important steps in daily clinical care.

The advantages of Gaston are mainly twofold: firstly, as a guideline representation model, Gaston facilitates the translation of paper-based clinical rules to executable models. By modeling the primitives given in Gaston editor and specifying each primitive through connections with knowledge components, the representation of clinical rules in Gaston is quite concise and fast to learn. Secondly, as a clinical decision support system, Gaston can automatically track the condition of a patient (e.g. whether the patient has certain allergies, whether the patient has taken certain medication or test), and thus to provide corresponding healthcare instructions and reminders to clinicians.

There are also some disadvantages of Gaston. First of all, currently Gaston is started by clinicians' manually click on a button of the interface in the hospital information system. There exists the risk that clinicians forget to trigger Gaston and thus the warning information would be missed. Second disadvantage is that Gaston has been applied to several areas in Catherina hospital, but its practical functions are limited to provide warnings and give instructions, more functions like timed warning and assisting in diagnoses can be further developed.

From the use perspective, users will not need much prior knowledge of using the Gaston decision support system during execution. Since Gaston will present warning messages or choice interface for users to choose and proceed. To model clinical rules in Gaston editor, users will need prior knowledge of modeling, thus medical experts can cooperate with experts of Gaston to develop the rule models.

To sum up, Gaston as a guideline-based clinical decision support system has facilitated clinicians' work by translating paper-based clinical guideline into computer interpretable guidelines that can be executed by Gaston as well as by providing warnings to them thus to improve the quality of clinical care. However, its functions are fixed and limited. More functions can be further developed in Gaston to expand its application fields.

6.4. Practical comparisons

After the descriptions and analyses of implementations of the anti-coagulation clinical guideline in YAWL, Declare and Gaston, now it comes to an integral practical comparison of all the systems. The practical comparison will refer to the theoretical analyses in chapter four, and make reflections on the dimensions of conceptual comparisons. In the last part, discussions on the possibilities of further improvement of the systems and better support of the anti-coagulation clinical guideline will be made.

The practical comparisons can be summarized in table 9 as below, followed by detailed illustrations of the table.

Systems	System functions	Advantages	Disadvantages	Use perspective
YAWL	Coordinate tasks, departments and people, enhance work efficiency	Reduce unnecessary communication time; Timed task to notify work to-do	Execution of task requires full knowledge of user; Time function insufficiency; Not sufficient to model	Required basic knowledge of Unified Modeling Language (UML) and XQuery; Complex process

			every exceptional situations	configuration and data definition
Declare	Coordinate tasks and people while at the same time provide extremely flexible support	Full support to the processes of exceptional situations in clinical environment	Data types are limited and simple; manual execution for every task; processes are too much simplified by Declare; unfriendly user interface	Easier to start with; clinical experts can build model by themselves in Declare
Gaston	Provide warnings to clinicians, notify critical issues	Facilitates the translation of paper-based clinical rules to executable models; track the condition of patient and give reminders	Manual trigger the start of Gaston; limited function to apply to other clinical fields	Easy to use: click a button then it will display information; modeling knowledge is required in building models in Gaston

Table 9 Practical comparisons of the systems

Workflow management systems are powerful tools in coordination of tasks and roles. The main objective of applying workflow management systems in an organization is to standardize processes in order to improve work efficiency. As in the practical implementation of anti-coagulation guideline, the biggest advantage of YAWL and Declare in supporting the clinical process is to minimize the waste of time on administrations and communications between departments, since they are flow-support systems, their main focus is process-centric coordination. Also by modeling clinical activities as workflow tasks in YAWL and Declare, the clinical activities will be displayed in the worklist panel in the systems, thus the systems can function as reminders to the clinicians. Both YAWL and Declare can be used for supporting processes in other domains, as long as the processes are manipulated by human not by machines and can be built as workflow models in the systems. This is in accord with the statement in section 5.1.1 that they are domain independence systems. The implementation in Gaston has provided distinct functions from in workflow management systems. The main function of Gaston in supporting the anti-coagulation guideline is to instruct or remind clinicians with specific content of clinical tasks by popping up warnings, thus the accuracy and safety of clinical activities and be assured. Gaston is a patient-centric or data-centric system that concentrates on patient data but not on organizational workflows. Moreover, the focus of Gaston system is more on the translation of paper-based clinical guidelines into executable clinical rules than executing them. This is quite different from the functions of workflow management systems. Since the modeling of guidelines in Gaston depends on primitives and PSMs, none of them is domain dependent; moreover, PSM can refer to knowledge components of various domains. Therefore, Gaston is a domain independent clinical decision support system.

From a lifecycle perspective, the use of YAWL and Declare starts from the planning phase, where roles and main tasks in the processes are planned before design in the editors, the previous phases of identification, discovery and diagnosis are left out since the clinical guideline flow chart has covered them. Also because all the implementations are in a lab environment, the deployment phase is excluded. So after the design phase, it comes to the execution phase, followed by control phase, diagnosis phase and re-design phase, etc. The procedures are in accord with the lifecycle circulation as in figure 18 of section 5.1.3. The starting phase of Gaston is basically from translating clinical experience or paper-based guidelines into clinical rules and then to build the rules in Gaston editor. Since the clinical rules in the anti-coagulation guideline are known in advance, the models can be built directly in Gaston and be tested. If the result is not in expectation, then refinement of clinical rules is needed to be made until the result is satisfactory. The steps match the lifecycle figure of Gaston in Figure 19 of section 5.1.3.

The representation of the anti-coagulation guideline in YAWL is a well-structured process model based on workflow patterns while in Declare is a loosely-structured process model based on declarative workflows. In the anti-coagulation clinical guideline, the data definitions and process specifications are explicit and more complex in YAWL. In Gaston, not the whole processes are modeled but only the clinical rules on a scenario or decision point that needs support is modeled. It is similar to a “decision tree” structure and is distinct from the support of workflow management systems where the processes are under comprehensive charge. In Gaston, guidelines are represented by its own framework (Figure 11, section 4.2) which consist of domain ontology (e.g. Knowledge component) and method ontologies (e.g. primitives and PSMs). The flowchart that modeled by primitives is explicitly defined: it contains steps, routes and the choice decision specified on some of the routes. On the contrary, the PSMs in the method ontologies implicitly describe knowledge and are used to refer to the knowledge components. Since Gaston only models the clinical rules which actually are not part of the organizational processes, the comparisons of it and the workflow management systems seem not to be on the same level. Thus YAWL and Declare are experimented to model the same rules and to compare with Gaston. This is not the other way around because Gaston is not able to integrate roles and to coordinate tasks. In YAWL, the three rules are modeled (Figure 43); the first task is for data input where users need to enter patient data and medication information, the next three parallel tasks will check the medication information and automatically redirect to the corresponding next task (notify patient or do nothing), then the process ends. It seems that YAWL can also support the clinical rules, however, as mentioned in the disadvantages before, YAWL cannot provide specific content of the task to users, users can only see “notify patient” or “do nothing” task in their worklist but without knowing where they come from. Declare is not able to model clinical rules since it doesn’t have decision task.

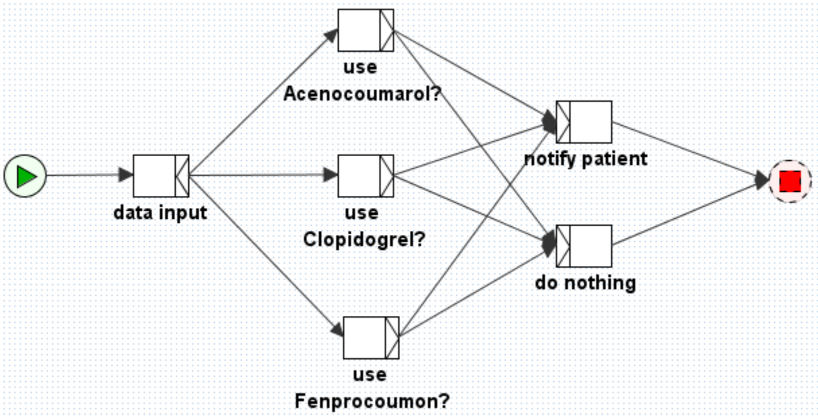


Figure 43: Modeling of clinical rules in YAWL

From the previous chapters, we already know that YAWL is good at supporting structured processes while Declare is best for providing extreme support to loosely-structured processes. But since the anti-coagulation clinical processes are partly structured and partly belonging to acute care, a combination of two systems is likely to provide better support. In this case, Declare will be used for modeling the top level of the processes, which contains several aggregations of processes. For example (Figure 44), the anti-coagulation processes can be divided into "administration", "PPOS", "medication check", "INR test", "medication control" and "intensive care" six sets, each of them contains structured and repetitive clinical processes that can be better support by YAWL. The INR test processes of YAWL as in Figure 23 can be modeled in a top-level aggregation task of "INR test" in Declare. In the "medication check" set, Gaston can be integrated as decision support system for providing warnings and specific instructions to clinicians. Therefore, during execution, clinicians will first choose and click on an aggregation task to start with from the Declare interface. Since the aggregation task in Declare is linked to YAWL processes, in the next step YAWL will be triggered and executed. The execution of Gaston can be triggered from task both in YAWL and Declare. For instance, the task "medication review" in the main process of YAWL (Figure 20) can be linked to Gaston automatically, thus after the previous task "planning operation" is executed, Gaston will start immediately and give corresponding warnings or messages to the clinicians. In Declare, a separate task can be defined as a trigger to start Gaston, in this way, the triggering of Gaston can be more flexible and convenient.

From a technical perspective, workflow systems can orchestrate and start other applications such as webmail service, spreadsheets, ERP systems, etc. This capability makes them ideal for implementing workflows involving systems and applications (Cardoso et al., 2004). Thus it's also probable that the three systems can be integrated. This possibility can be further explored and examined in future researches.

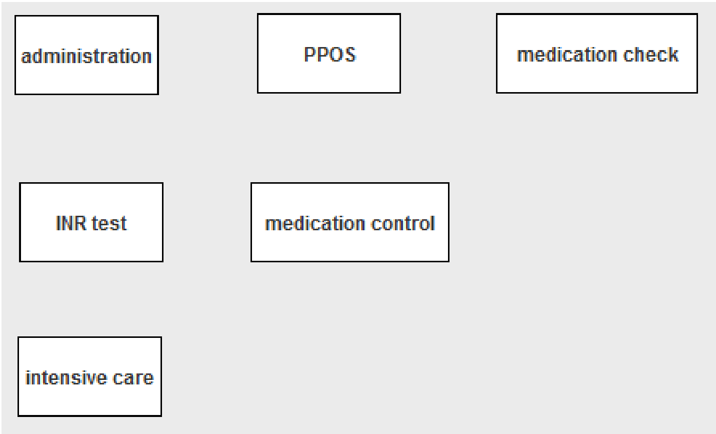


Figure 44: Aggregation of tasks in Declare

In terms of the possibilities of improvement for those systems, since YAWL and Declare are both academic workflow management systems, especially Declare is a prototype of constraint-based workflow management system, the user experience perspectives have great potential to be improved. Moreover, both YAWL and Declare have bugs in some of functions. The bugs in YAWL are not critical and won't impact normal use but still are not user friendly. For instance, the data created in net variables in YAWL can only be changed but cannot be deleted. This causes the consequence that an empty or wrong data exists in the net variable and so that the validation tool will always detect mistakes. Thus the only solution is to transform the data into a new variable. In Declare, the bugs are even more critical. The execution of Declare is influenced by the bugs not always run successfully. In terms of Gaston, since it's already a mature system used by the hospital, there are no bugs that would influence its functions. But there are still some perspectives that can be improved: first is that a time trigger can be developed in the execution of clinical rules by Gaston so that it won't be always manually triggered and thus clinical errors can be further reduced; second perspective is that Gaston should be used for translating more clinical rules and providing more support in the clinical environment since currently it has only been applied in limited areas.

To sum up, YAWL and Declare as two different types of workflow management systems and Gaston as a guideline-based clinical decision support system are quite different in terms of their application goal, structure but have some common functionalities. Each system has its own expertise and also insufficiencies. But since some of the insufficiencies of one system can be compensated by the expertise of one other system, it is proposed to combine the three systems in the anti-coagulation clinical process so as to achieve better support. This proposal can be further examined by researchers in future work.

7. Conclusion

The main aim of this study is to investigate the functions of workflow management systems (WFMSs) and clinical decision support systems (CDSSs) in clinical environment, the advantages and disadvantages of both the systems, whether they can replace each other in some fields, and the possible improvement of the two types of systems. One hypothesis is proposed at the beginning of the study: The functions of WFMSs and CDSSs are partly overlapped; they both have their own properties and features that cannot be replaced by each other. The study is conducted from two folds: theoretical analyses of the three WFMSs (YAWL, FLOWer and Declare) and one CDSS (Gaston) by comparing previous researches and practical analyses of the clinical guideline executions in YAWL, Declare and Gaston.

The theoretical analyses are made based on four perspectives: domain scope, technological scope, lifecycle scope and knowledge translation. In the conceptual comparisons, YAWL, FLOWer and Declare as flexible workflow management systems are found to be distinct from Gaston as a guideline-based clinical decision support system in several areas, especially from their main application goal, type of support, guideline representation model and the modeling languages. To validate the conceptual comparisons, practical implementations of the anti-coagulation clinical guideline are conducted in those systems. Workflow models are built in YAWL and Declare while the clinical rules extract from the guideline are built in Gaston. As a result, YAWL and Declare have played the role of a coordinator of tasks and roles as expected from the theoretical analyses whereas Gaston plays the role of warning system. Although the clinical rules can also be built in YAWL, however, YAWL is not as competent as Gaston in decision support.

So far, the hypothesis of the project has been partly verified. In short, YAWL as workflow management system can support clinical rules to a moderate extent, but this requires the clinicians to have full knowledge of the full content of the task in execution. But in the other way around, Gaston as a clinical decision support system is not able to support organizational workflows due to the fact that it has no integration of resources. But in order to better support the anti-coagulation clinical guideline, it is proposed to integrate YAWL, Declare and Gaston, where Declare focuses on coordination of task aggregations in high level, YAWL is responsible for modeling structured and repetitive processes in each task aggregation and Gaston provides decision support for the clinical rules in the process. But this possibility is required further examination, especially from a technical perspective.

There are also several limitations in the project. First of all, the number of workflow management systems and clinical decision support systems that included in the analyses of the project is limited. Other workflow management systems and non-knowledge-based clinical decision support systems can be included for further investigation. Second, the implementations of clinical guidelines are under laboratory environment, thus many factors in hospital environment are excluded, such as technical problems with integrating hospital information systems, etc. This is beneficial on one hand since all the systems are compared in the same condition, but on the other hand, results obtained from the hospital environment will be more accurate. Future research can implement the clinical guidelines in hospital environment, but the possibility of the systems to be integrated in the hospital information system should be firstly considered. Also the legality of using real patient data and the risk of leak patient information should also be taken into account. Third, as clinical processes can be categorized into several types and the study only examines one clinical guideline, thus the other types of clinical guidelines can be modeled and tested in future research.

References

Alonso, G ., Agrawal, D ., El Abbadi, A ., & Mohan, C . (1997). Functionality and Limitations of Current Workflow Management Systems: IBM Almaden Research Center.

Berner, Eta S., & Lande, Tonya J. (2007). Overview of Clinical Decision Support Systems

Clinical Decision Support Systems. In E. S. Berner (Ed.), (pp. 3-22): Springer New York.

Cardoso, Jorge, Bostrom, Robert P., & Sheth, Amit. (2004). Workflow Management Systems and ERP Systems: Differences, Commonalities, and Applications. *Information Technology and Management, 5*(3), 319-338. doi: 10.1023/B:ITEM.0000031584.14039.99

De Clercq, P. A., Blom, J. A., Hasman, A., & Korsten, H. H. M. (2000). GASTON: an architecture for the acquisition and execution of clinical guideline-application tasks. *Medical Informatics & the Internet in Medicine, 25*(4), 247-263.

Dwivedi, A., Bali, R. K., James, A. E., & Naguib, R. N. G. (2001, 2001). *Workflow management systems: the healthcare technology of the future?* Paper presented at the Engineering in Medicine and Biology Society, 2001. Proceedings of the 23rd Annual International Conference of the IEEE.

Grossman, J.H., & Field, M.J. (1990). *Clinical practice guidelines: directions for a new program* (Vol. 90): National Academies Press.

Institute of Medicine. (2012). Retrieved from http://www.iom.edu/

Ledley, R. S., & Lusted, L. B. (1959). The Use of Electronic Computers to Aid in Medical Diagnosis. *Proceedings of the IRE, 47*(11), 1970-1977. doi: 10.1109/JRPROC.1959.287213

Lenz, R., Peleg, M., & Reichert, M. (2012). *Healthcare Process Support: Achievements, Challenge, Current Research*.

Mans, Ronny, Van der Aalst, Wil, Russell, Nick, & Bakker, Piet. (2009). Flexibility Schemes for Workflow Management Systems

Business Process Management Workshops. In D. Ardagna, M. Mecella & J. Yang (Eds.), (Vol. 17, pp. 361-372): Springer Berlin Heidelberg.

Mans, Ronny, Van der Aalst, Wil, Russell, Nick, Bakker, Piet, Moleman, Arnold, Lassen, Kristian, & Jørgensen, Jens. (2009). From Requirements via Colored Workflow Nets to an Implementation in Several Workflow Systems

Transactions on Petri Nets and Other Models of Concurrency III. In K. Jensen, J. Billington & M. Koutny (Eds.), (Vol. 5800, pp. 25-49): Springer Berlin / Heidelberg.

Perreault, L., & Metzger, J. (1999). A pragmatic framework for understanding clinical decision support. *Journal of Healthcare Information Management, 13*(2), 5-21.

Pesic, M., Schonenberg, H., & Van der Aalst, W. M. P. (2007, 15-19 Oct. 2007). *DECLARE: Full Support for Loosely-Structured Processes.* Paper presented at the Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International.

Reijers, H. A., Rigter, J. H. M., & van der Aalst, W. M. P. (2003). The Case Handling Case. *International Journal of Cooperative Information Systems, 12*(03), 365-391. doi: doi:10.1142/S0218843003000784

Schonenberg, Helen, Mans, Ronny, Russell, Nick, Mulyar, Nataliya, & Aalst, Wil. (2008). Process Flexibility: A Survey of Contemporary Approaches

Advances in Enterprise Engineering I. In J. L. G. Dietz, A. Albani & J. Barjis (Eds.), (Vol. 10, pp. 16-30): Springer Berlin Heidelberg.

Terenziani, Paolo. (2010). A Hybrid Multi-layered Approach to the Integration of Workflow and Clinical

Guideline Approaches

Business Process Management Workshops. In S. Rinderle-Ma, S. Sadiq & F. Leymann (Eds.), (Vol. 43, pp. 539-544): Springer Berlin Heidelberg.

Van der Aalst, Wil, Aldred, Lachlan, Dumas, Marlon, & Ter Hofstede, Arthur. (2004). Design and Implementation of the YAWL System

Advanced Information Systems Engineering. In A. Persson & J. Stirna (Eds.), (Vol. 3084, pp. 281-305): Springer Berlin / Heidelberg.

Van der Aalst, Wil, Basten, T., Verbeek, H. M.W., Verkoulen, P. A.C., & Voorhoeve, M. (1999). *Adaptive Workflow.* Paper presented at the Filipe (Ed.), Enterprise Information Systems, Norwell.

Van der Aalst, Wil, Pesic, M., & Schonenberg, H. (2009). Declarative workflows: Balancing between flexibility and support. *Computer Science - Research and Development, 23*(2), 99-113. doi: 10.1007/s00450-009-0057-9

Van der Aalst, Wil, & Ter Hofstede, A. H. M. (2005). YAWL: yet another workflow language. *Information Systems, 30*(4), 245-275. doi: 10.1016/j.is.2004.02.002

Van der Aalst, Wil, & Van Hee, K. (1997). *Workflow Management: Models, Methods, and Systems*. Massachusetts London: The MIT Press Cambridge.

Van der Aalst, Wil, Weske, Mathias, & Grünbauer, Dolf. (2005). Case handling: a new paradigm for business process support. *Data & Knowledge Engineering, 53*(2), 129-162. doi: 10.1016/j.datak.2004.07.003

Wang, D., Peleg, M., Tu, S. W., Boxwala, A. A., Greenes, R. A., Patel, V. L., & Shortliffe, E. H. (2002). Representation primitives process models and patient data in computer-interpretable clinical practice guidelines: a literature review of guideline representation models. *Int J Med Inform, 68*(1-3), 59-70.

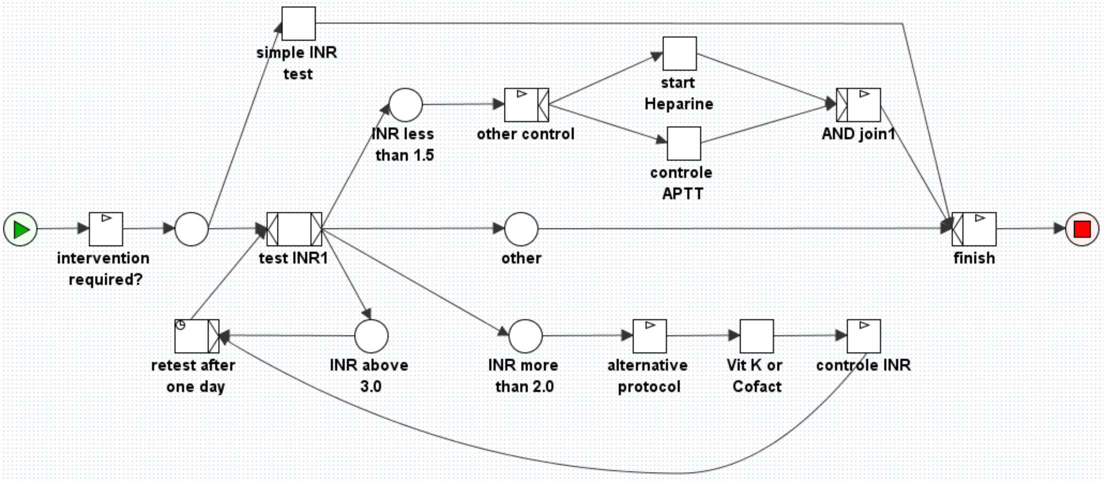
Workflow Management Coalition. (2012). Retrieved from http://www.wfmc.org/

Wyatt, J., & Spiegelhalter, D. (1991). Field trials of medical decision-aids: potential problems and solutions. *Proc Annu Symp Comput Appl Med Care*, 3-7.

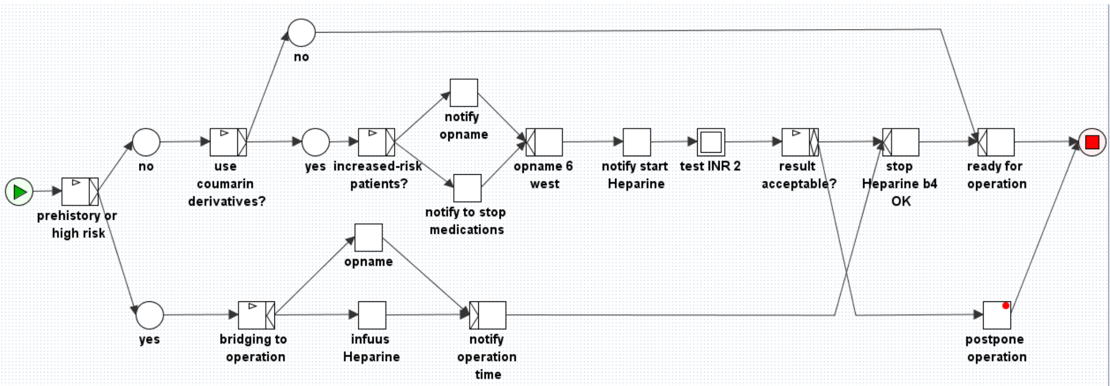
Appendix

Appendix A --- YAWL process model of anti-coagulation clinical guideline

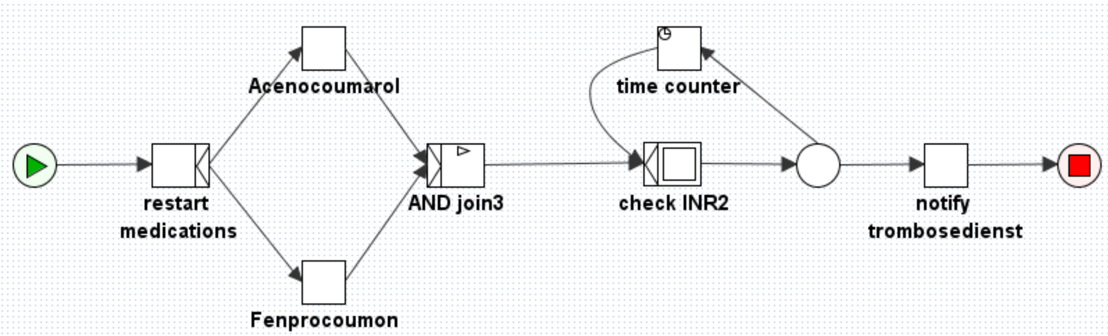
- **INR Test**



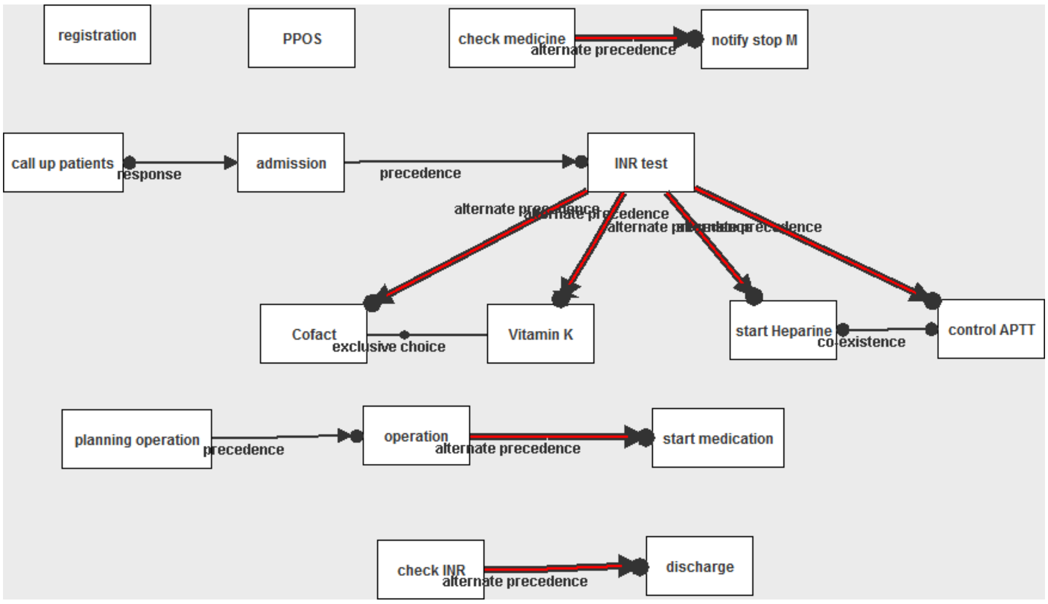
- **Therapy**



- **Intensive Care**

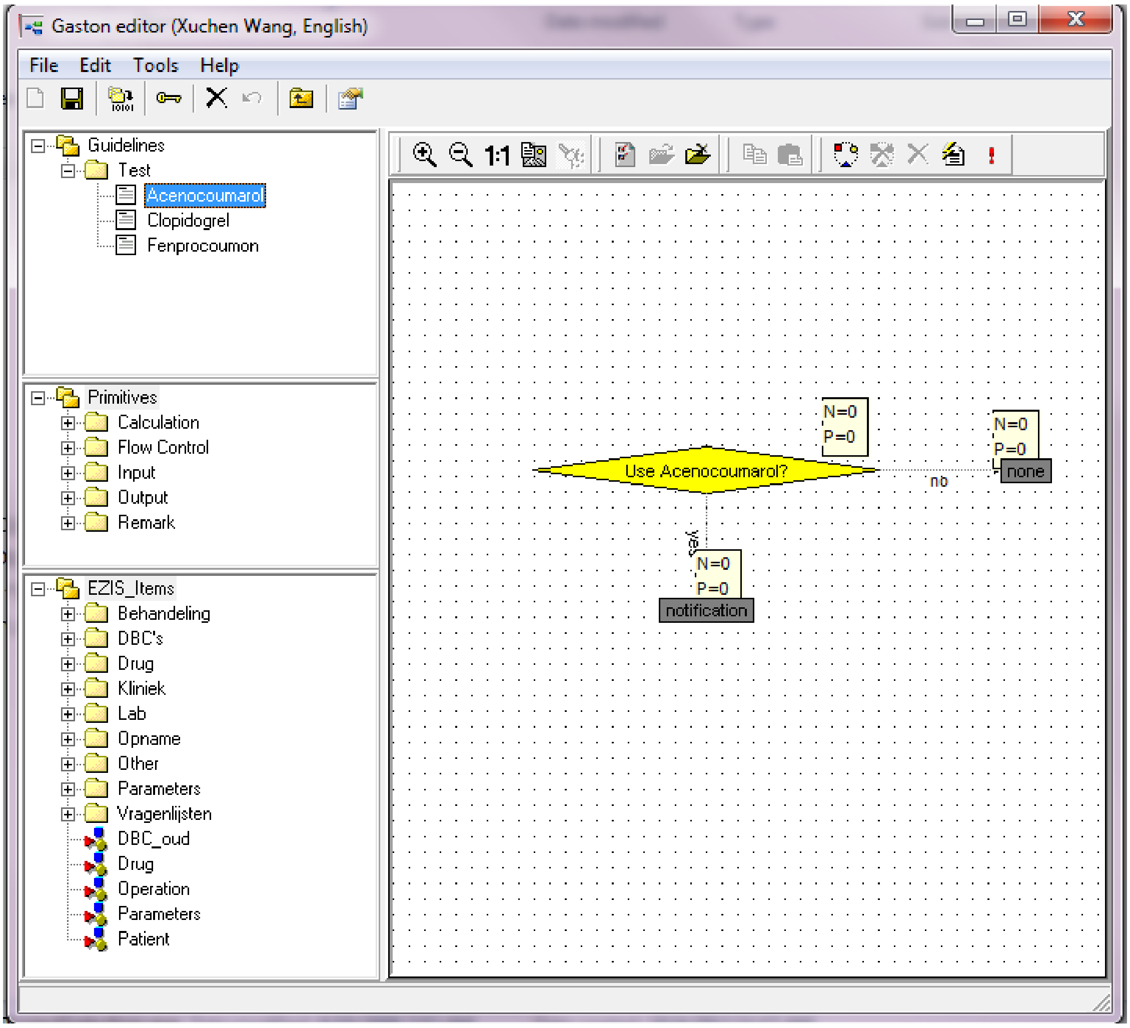


Appendix B --- Declare process model of anti-coagulation clinical guideline

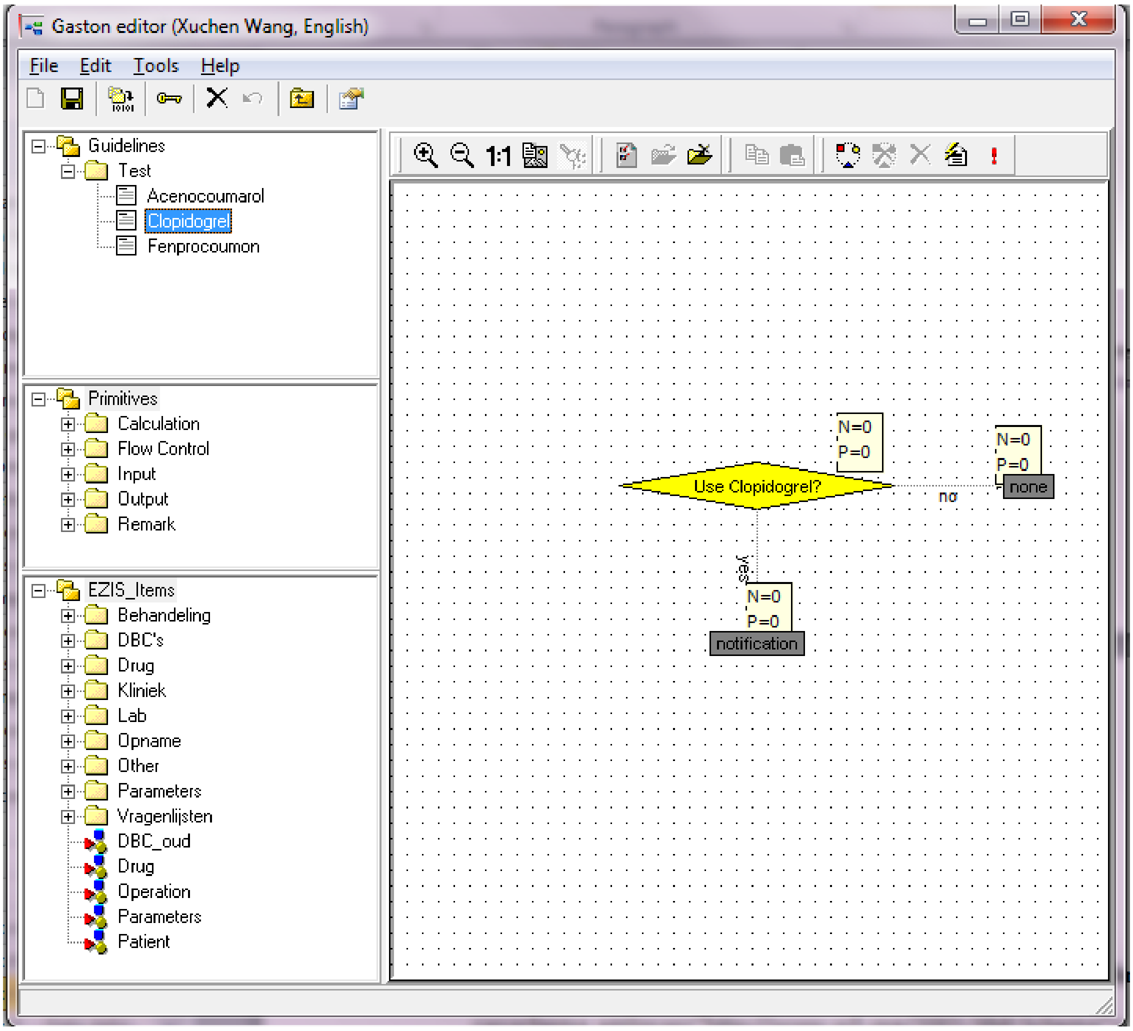


Appendix C --- Gaston model of anti-coagulation clinical guideline

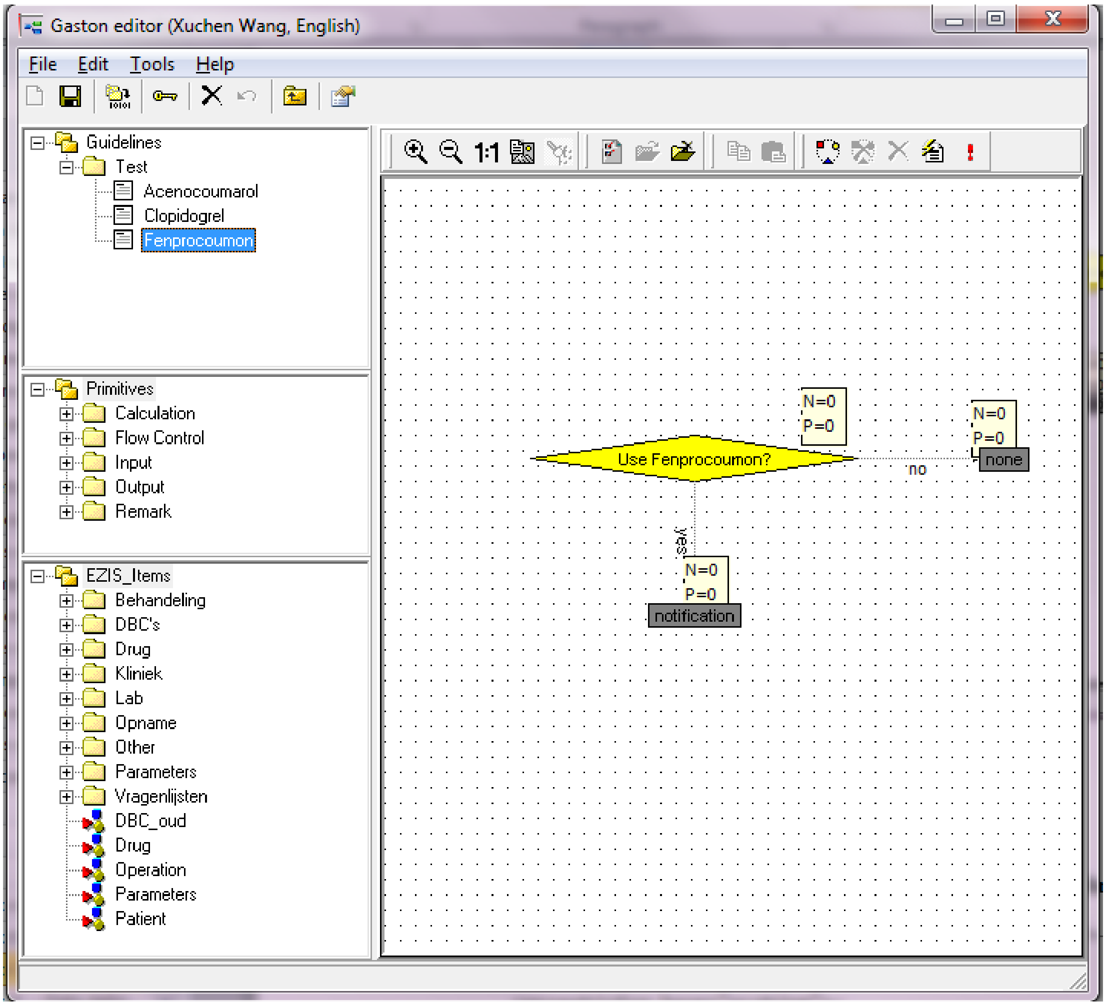
- Acenocoumarol



- Clopidogrel



- Fenprocoumon



Appendix D ---Data definition of anti-coagulation clinical guideline in YAWL

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:simpleType name="patientType">
        <xs:restriction base="xs:string">
            <xs:enumeration value="PPOS" />
            <xs:enumeration value="Overname" />
        </xs:restriction>
    </xs:simpleType>
    <xs:simpleType name="patientRisk">
        <xs:restriction base="xs:string">
            <xs:enumeration value="HIGH" />
            <xs:enumeration value="LOW" />
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="patientInfo">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="PatientID" type="xs:long" />
            <xs:element maxOccurs="1" minOccurs="1" name="PatientName" type="xs:string"
/>
```

```
            <xs:element maxOccurs="1" minOccurs="1" name="Age" type="xs:integer" />
            <xs:element maxOccurs="1" minOccurs="1" name="Gender">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value="Male" />
                  <xs:enumeration value="Female" />
                </xs:restriction>
              </xs:simpleType>
            </xs:element>
            <xs:element maxOccurs="1" minOccurs="1" name="Address" type="xs:string" />
            <xs:element maxOccurs="1" minOccurs="1" name="PhoneNumber" type="xs:string"
/>
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="prehistory">
        <xs:restriction base="xs:string">
            <xs:enumeration value="AVRorMVR" />
            <xs:enumeration value="NONE" />
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="patient">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="PatientInfo" type="patientInfo"
/>
            <xs:element maxOccurs="1" minOccurs="1" name="PatientType"
type="patientType" />
            <xs:element maxOccurs="1" minOccurs="1" name="Prehistory" type="prehistory" />
            <xs:element maxOccurs="1" minOccurs="1" name="PatientRisk" type="patientRisk"
/>
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="medicationInfo">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="MedID" type="xs:long" />
            <xs:element maxOccurs="1" minOccurs="1" name="Name" type="xs:string" />
            <xs:element maxOccurs="1" minOccurs="1" name="Dosage" type="xs:string" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="subscription">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="SubscriptionTime"
type="xs:time" />
            <xs:element maxOccurs="5" minOccurs="1" name="MedInfo"
type="medicationInfo" />
```

```
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="operation">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="DoctorName" type="xs:string"
/>
            <xs:element maxOccurs="1" minOccurs="1" name="OperationTime"
type="xs:dateTime" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="screening">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="DoctorName" type="xs:string"
/>
            <xs:element maxOccurs="1" minOccurs="1" name="ScreeningTime"
type="xs:dateTime" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="patientNotification">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="SendNotification"
type="xs:boolean" />
        </xs:sequence>
    </xs:complexType>
    <xs:simpleType name="patientCondition">
        <xs:restriction base="xs:string">
            <xs:enumeration value="OK" />
            <xs:enumeration value="NOK" />
        </xs:restriction>
    </xs:simpleType>
    <xs:complexType name="INRLevel">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="testTime" type="xs:time" />
            <xs:element maxOccurs="1" minOccurs="1" name="INRLevel" type="xs:decimal" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="medicationAction">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="MedicationAction"
type="xs:string" />
            <xs:element maxOccurs="1" minOccurs="1" name="StartTime" type="xs:time" />
            <xs:element maxOccurs="1" minOccurs="1" name="Executed" type="xs:boolean" />
        </xs:sequence>
    </xs:complexType>
```

```
    <xs:complexType name="medicationChecklist">
        <xs:sequence>
            <xs:element maxOccurs="10" minOccurs="1" name="MedicationChecklist"
type="medicationAction" />
        </xs:sequence>
    </xs:complexType>
    <xs:complexType name="postponeOK">
        <xs:sequence>
            <xs:element maxOccurs="1" minOccurs="1" name="PostponeOK" type="xs:boolean"
/>
        </xs:sequence>
    </xs:complexType>
</xs:schema>
```