2015

Clinical artefact networks

Katayoun Khodaei
University of Wollongong

University of Wollongong

CLINICAL ARTEFACT NETWORKS

A Thesis Submitted in Partial Fulfilment of
the Requirements for the Award of the Degree of

Master of Computer Science

from

UNIVERSITY OF WOLLONGONG

by

Katayoun Khodaei

School of Computer Science and Software Engineering
Faculty of Informatics

2015

CERTIFICATION

I, Katayoun Khodaei, declare that this thesis, submitted in partial fulfilment of the requirements for the award of Master of Computer Science, in the School of Computer Science and Software Engineering, Faculty of Informatics, University of Wollongong, is wholly my own work unless otherwise referenced or acknowledged. The document has not been submitted for qualifications at any other academic institution.

(Signature Required)

Katayoun Khodaei
30 March 2015

Contents

List of Tables

List of Figures

ABSTRACT

A major challenge for healthcare systems is to manage the continuously increasing amount of disparate data. Such large quantities of unstructured data, spread across multiple data sources, makes it difficult and time consuming for clinicians to locate the information they need to perform their tasks efficiently and accurately. This makes it difficult for knowledge workers to identify the relevant information that they need to perform their daily tasks. The ability to retrieve this information is constrained by not only the sheer volume of data but also the logical disconnection between the physical storage of the data and the processes that created the data as, in most situations, clinical processes and process related data are managed separately. It is of prime importance not only in clinical decision making but also for the safety of the patients for clinicians to be able to retrieve the appropriate information, in the appropriate level of granularity, in a timely manner. Clinical artefact networks (CA-Nets) are introduced with the aim of addressing this issue by representing the existing data in a contextual format. With CA-Nets, we aim to correlate the semantic aspect of the data irrespective of how, where and in what format the data are stored. The resulting model will then provide an ability to navigate through the collection of data items as one would navigate through a graph or network. Such a model will provide the appropriate information at the required level of granularity. CA-Nets are developed based on semantic networks and data provenance techniques. Thus, this thesis addresses an important problem in supporting clinical decisions-that of generating and representing contextual knowledge.

KEYWORDS: CA-Nets, Knowledge Representation, Business Process Management, Semantic Networks, Provenance Management

Acknowledgements

I would like to thank my supervisor Prof. Aditya Ghose, my second supervisor Dr.Chee-Fon Chang for their kind support and help. Thank you to all of the DSL members for their kind help. Thank you to Dr Andrew Miller for his time and kind assistance. Thank you to Evan Morrison for his kind support and help.

Chapter 1

Introduction

These days one of the major challenges for healthcare professionals is to deal with the continuously increasing amount of disparate data [30, 44]. Such large quantities of unstructured data, spread across multiple data sources, makes it difficult and time consuming for clinicians to locate the information they need to perform their daily tasks efficiently and accurately. It is of prime importance not only in clinical decision making but also for the safety of the patients for clinicians to be able to retrieve the appropriate information, in the appropriate level of granularity, in a timely manner. The ability to retrieve this information is constrained by not only the sheer volume of data but also the logical disconnection between the physical storage of the data and the processes that created the data as, in most situations, clinical processes and process related data are managed separately [44].

1.1 Research Objective

This study proposes a conceptual model which is called CA-Nets. With CA-Nets, this study aims to correlate the semantic aspect of the data irrespective of how, where and in what format the data are stored. The resulting model will then provide an ability to navigate through the collection of data items as one would navigate through a graph or network.

Such a model will provide the appropriate information at the required level of granularity. With this capability, the model is then in a position to answer not only a range of queries but also provide the machinery to help justify clinical decisions. To achieve such goal, the model will have to be generated dynamically as queries to a network that are generally context specific. For example, when an oncologist reviews a patients record, the specialist is only interested in the information related to the cancer disease and not necessarily other illnesses that the patient had in the past.

CA-Nets are knowledge representation tool which represent the information about the patient specific treatment. The information is represented in a structure which helps user to gain required knowledge. Thus, this study uses the term knowledge when discussing about represnting information based on specific rules and structures.

1.2 Research Methodology

CA-Nets uses several technologies and technique as its fundamental machinery. Semantic networks [4] forms the fundamental basis to correspond activities and their outcomes. This provides the basis for the knowledge representation aspect of CA-Net. Data provenance [18], is used to provides complementary metadata about the acquiring history of the data product, including input data sources transformed to generate. In healthcare context, data provenance captures the evolution of patient medical history.

For the purpose of defining an appropriate motivation example for this study, several meetings were held with one of the radiation oncologist from a local cancer care center. Based on the gathered information and requirement analysis, a suitable motivation example is defined to use through the whole study. This helps the basis for developing and explaining the research study. Moreover, to represent the use of the CA-Nets, several meetings were held in the local cancer care center to observe the process of chemotherapy protocols which will be explained in detail in chapter 5.

1.3 Research question

This research addresses two research questions. Firstly, given the volume of disparate unstructured data in a range of data stores, how can we extract the required knowledge from these readily available sources? Secondly, how can we represent knowledge about the patient specific treatment in a manner that supports a range of clinical queries?

1.4 Research Structure

This thesis is organised as follows: chapter 2 presents the background knowledge covering related works such as clinical process management and knowledge representation, provenance management, semantic annotation and abductive reasoning. Chapter 3 provides the formal definition of CA-Nets, and explain its use with examples. Chapter 4 presents a detailed discussion on the use of CA-Nets as well as a comparison between CA-Nets and a related framework; Process-oriented Information Logistics(POIL). Chapter 5 illustrates the practical use of CA-Nets with two scenarios and finally chapter 6 concludes the research.

Chapter 2

Literature Review

2.1 Introduction

In the following sections, we will firstly introduce workflow management systems [31] as background for discussing careflow management systems. We reviewed the current major issues within care flow management systems. Process mining is discussed in detail, because it is used for implementing our proposed model. We are providing a review on semantic networks [4, 55, 65] because it forms the fundamental basis for the proposed framework(CA-Nets). Provenance management [7, 18, 20, 63] is reviewed as we use this technique in our approach. Semantic annotation [23] and abductive reasoning theory [19, 34] are briefly discussed as they form the fundamental technique for reasoning in our approach. Finally, we present a discussion on process-oriented information logistics (POIL) [29]. This discussion is particularly important as forms the basis for a comparison with CA-Nets.

2.2 Clinical Process Management

For health care organisations, the strategic goal today is to improve the quality of care [2]. Good quality of care means providing patients with competent services in a technically ap-

propriate manner, with good level of communication, shared decision making, and allowing a fair amount of flexibility to deal with unexpected situations [2]. Business process management (BPM) is vital for any businesses to assist them with identifying needs, to wisely allocate resources, to reduce errors, monitor the processes and verify the system outcomes [75]. This study specifies the business to health care providers, thus it uses clinical process management instead. Clinical process management(CPM) is defined as "Supporting clinical processes using methods, techniques, and software to design, enact,control, and analyse operational processes involving humans, organisations, applications, documents and other sources of information" [82]. CPM can be considered as an extension of classical Workflow Management (WFM) systems and approaches [75]. The Workflow Management Coalition [31] provides the following definition: "A workflow consists of the automated business processes in whole or part, where documents, information and tasks are passed from one participant to another based on the predefined rules in order to achieve the overall goal".

Workflow Managment Systems (WfMSs) are widely used to identify needs, to decrease the errors and duplication of work, to ensure completion of processes on time and in accordance with defined rules, and to ease documentation [46]. A WfMS, can be viewed as a knowledge-based system which assigns the right task to the right person at the right time. The workflow concept is closely related to the notion of processes in industrial and office environments. A process definition comprises a network of activity steps related to human or computer operations and rules to monitor the progression through the activities [31].

2.2.1 Current Process and Information Management in Health Care Systems

2.2.1.1 Careflow Management

Workflow management systems designed and implemented for the healthcare domains are called careflow management systems (CfMSs). CfMSs are case based, i.e. an instance of

the workflow is the execution of the care process for any specific patient [10, 62]. One can think of a patient care process as a careflow instance. The aim of a CfMS is to manage patients by executing clinical tasks in a particular order. A careflow process definition specifies which tasks require to be executed and in what order. In other words, the coordinated execution of multiple medical tasks performed by various healthcare subjects for a particular patient [25]. The advantages of this approach are that it flows the work more autonomously and facilitates the exchange of case-related information and also assists users in monitoring the progress of whole tasks [25]. A patient's journey through the health care system can be improved by CfMS integration. CfMSs can be used in a variety of ways, as decision support systems that aid care providers in optimally diagnosing and treating patients [80], as control mechanisms to monitor or detect co-incident treatment process conflicts [28], as an analyser for monitoring and identifying main causes of errors to support patient safety [15, 59].

There exist process management systems which are designed with the aim of optimising CfMSs. ADEPT, was a project conducted by Dadam and Reichert [16] which designed as a process management system with the aim of ease of use and the support of high flexibility through process change. The Architecture of Integrated Information Systems (ARIS) [39], which is a business process architecture developed by IDS Scheer AG has been applied in a variety of industrial applications. This framework has been applied by Leu and Huang [39] to optimise the clinical processes of an emergency department. Process optimisation was also considered by Bürkel, Baur, and Höss [8] using two methods namely, "rapid prototyping" of a clinical workflow and a two level approach using round robin and individual semi-structured interviews to focus upon areas of the workflow which may be optimised.

Careflow evaluation has recently attracted research attention. Borycki et al. [6] used simulation to evaluate a clinical workflow and system impact based on simulated user interactions. The approach [6]consists of data collection via audio and video recordings of interactions between healthcare workers and health care information systems. Methods

based on simulations have been used in areas such as biomedical informatics to study human computer interaction focusing on areas like: human factors, usability, doctor patient interactions, health professional decision making, medical error and new device testing.

2.2.1.2 Careflow Modelling and Implementation

Process modelling is the task of representing processes of an enterprise, so that the current ("as is") process may be analysed and improved in future ("to be") [13]. Various languages and tools have been used to represent workflows. The Kiepuszewski et al. in [35] have evaluated 15 different WfMSs and the results demonstrate that current products use a variety of workflow languages resulting in different capabilities. This motivated them to focus on the fundamentals in workflow for handling this issue. They have focused on control flow perspective, which defines the order and sequence of activities and their execution which allow flow of execution control, parallelism or synchronisation. Petri-nets have been identified as useful in modelling and analysing careflows [69]. The advantages of using Petri-nets lies in the combination of a mathematical foundation and a comprehensive graphical representation. During the previous two decades the classical Petri-nets have been enhanced with colour, time and hierarchy. These enhancements facilitate the modelling of complex processes where data and time are critical factors [71]. Yet Another Workflow Language (YAWL) [74] is another well-known language, which has been developed based on Petri-nets with the ability to allow for more direct and intuitive support of the workflow patterns with more complexity. This language was developed to enable users to map complex patterns like multiple instances, complex synchronisations or non-local withdrawals. Business Process Model and Notation (BPMN) [61] is an established standard for business process modelling in industry, which also has been used for clinical process modelling. BPMN is introduces with the aim of providing a notation that is understandable by both business users such as business analysts and the technical developer [13]. BPMN [61] is a flowcharting

technique for creating graphical models of business operation processes, which are generated into Business Process Diagrams. The notation consists of a set of graphical elements such as: activity objects, event objects, flow objects, connectivity objects, grouping objects, annotations and artefacts [75].

Gattnar and Ekinci [21] proposed a process-based quality measurement model to support and measure the effectiveness and efficeincy of quality of clinical care. For this purpose they describe a clinical reference process model using Event-driven Process Chains [77]. They integrated generic clinical time-based Key Performance Indicators, to measure the timeliness of acute care in standardised way. In this case process modelling starts with the commencement of patient's acute symptoms and finish with the patient's discharge from hospital. Huang, Zhu, and Wu [32] analysed the knowledge used in process modelling and designed the lifecycle of the process modelling. They offered a customer-centred careflow system to provide an efficient way to build the process model and achieve better reusability of the model.

As there has been a growing interest in web services, languages like BPEL (Business Process Execution Language for Web Services) for implementing processes based on web services, and DecSerFlow (Declarative Service Flow Language) for a declarative style with the ability to monitor web services, have been introduced [72, 73]. BPEL can be used to define a model and a grammar for describing the behaviour of a business process according to interaction amongst the processes. It can be considered one of the perfect choices to translate BPMN diagrams into executable code [13].

A great deal of effort has been put into the design, implementation and management of applicable CfMSs, but many of them do not meet the expectations of physicians or health care organisations a real working environment. For instance, in [52, 53] a careflow management system was developed for the post-stroke rehabilitation, but it didn't use in a real work setting. A CfMS implementation for the care of diabetes patients is defined by Pan-

zarasa et al. [51] with the focus on inter-organisational communication. The study doesn't mention an implementation in a clinical setting any or real-world constraints that may affect the use of the system in a hospital environment. An approach is defined by Maximini et Shaaf [42] to support the execution of structured workflows and knowledge-intensive tasks which cannot be described in advance and does not enforce physicians to follow a predefined methodology.

An overview regarding careflow management, modeling and implementation is provided as a background regarding the objective use of careflow in different area of healthcare systems. As part of our research we use careflow management systems using BPMN(as a common industrial standard language) to implement and represent CA-Nets use in healthcare management systems.

2.2.2 Key Issues within Process Management in Health Care Systems

2.2.2.1 Flexibility through Case handling and Adaptation

One of the most important issues in CfMSs is the lack of flexibility, which can be defined as the ability of the system to execute based on a loosely defined model which is completed at run-time and could be unique to each process instance [38]. Flexibility is the ability of system to cope with unspecified circumstances by inserting, dropping or even changing the sequence of the planned task execution. Reijers et al. [56] proposed a methodology for creating processes with the required degree of flexibility for deployment in the healthcare system to support automated flexibility in a CfMS.

Some researchers [52, 68] have proposed methodologies and developed systems able to deal with exceptions. Panzarasa et al. [52] define an exception as "any deviation from an ideal care delivery process that uses available resources to achieve the desired clinical goals in an optimal way". Exceptions can happen when the available resources or task priorities change or incorrect and late tasks performed [52]. Thom et al. [68] focus on the

real exceptions which happen in the chemotherapy treatment in a German hospital. The presented exceptions are discussed from different aspects such as medical errors, organisational guidelines, patient health status and technical contingencies. They analyse the nature of these exceptions and the way users deal with them in order to relate them to clinic exception handling patterns.

Two approaches exist which both attempt to make process management more flexible. The first one is case handling, which supports the problem of flexibility "by anticipating volatile environments and thus avoiding the need for process modification" [25]. This approach is evaluated by Mikolajczak, Shenoy and Shah [45, 62] based on a case study of cutaneous melanoma and gastric cancer, respectively. The other approach, adapting, helps flexibility handling by providing the means to modify the process definition during execution is the other approach [25]. The ability for dynamic adaptation of an in-progress workflow is a significant requirement for a CfMS [67]. Hani Tawil et al. [26] also discuss this issue by applying BPMN ontology to careflow modelling in order to assist model evolution and flow migration. Rinderle, Reichert and Dadam [57] compare some of the dynamic-adapting approaches and also address the correct adaptation of running workflows and analyse the actual approaches that satisfy them. AgentWork [47] uses dynamic rules to allow users to identify possible problems and the WfMS will adapt the workflow according to the predefined dynamic rules. Such rules are very useful because they allow the WfMS to cope with situations that are exceptional [47].

2.2.2.2 Verification

Formal verification is one of the growing fields which tend to formalise and verify specifications in systems. This method can assist in checking workflow execution as desired and to find the errors in the system [17]. Dallien et al. [17], focused on formal verification by checking workflows before they are put into action, with the finding that potential errors

were reduced or controlled. They developed a language for expressing guidelines and described a model checker implementation which is written in XSB Prolog. Chesani et al. [11] presented an algorithm capable of translating a careflow model to a formal language based on computational logic and abductive logic programming with the aim of verifying the conformance of a given careflow process execution according to the given careflow model. A web-based CfMS is proposed in which extends the ability of WfMS by taking advantage of web technology and formal verification features. Miller et al. [46] selected model checking as the verification method. This framework provides high performance computing methods to support real-time monitoring and adaptation, as well as allowing shared knowledge between collaborators by integrating ontologies.

2.2.2.3 Process Mining and Conformance Checking

In real situations, there is usually a significant gap between what actually happens and what is supposed to happen in healthcare systems. Process mining attempts to provide a concise assessment of reality which can be helpful in verifying process models and finally be used in the process redesign efforts [48]. Many organisations make use of a wide variety of Process-Aware Information Systems (PAISs) [48] to support their business processes. These systems commonly log events related to the actual business process executions, which can be used for different purposes. Process mining are used to extract process related information from process logs [40] (process discovery), and evaluate the adequacy of a model in describing a log (process conformance) [48]. Process mining is an area of research that is useful in the identification and analysis of formal models in a PAIS, in order to support its design and maintenance [48]. Mining processes has received a lot of attention, which is also reflected in the research efforts seen in this area [58]. Careflow mining techniques are considered as a way to deal with unstructured processes in [9]. By analysing the event log of a careflow process of a specific hospital, the researchers aimed to stimulate the de-

velopment and adoption of unstructured processes (spaghetti-like processes) and careflows. Mans et al. [40] used process mining to discover inconsistencies in a stroke guideline and the reasons for those inconsistencies. A study is conducted by Peleg et.al. [54], that was used to mine the processes at the semantic level to enhance healthcare processes. In this approach, healthcare process instance data is used to learn the best possible path needed in order to attain appropriate outcomes for patients with different characteristics.

The problem of obtaining a formal model from a log is known as process discovery [48]. Several algorithms exist which attempt to derive models by observing process traces in the log. α -algorithm is a well known algorithms and which is described in detail by van der Aalst [71]. This method has limited capability as it will only work based on a noise-free workflow log with sufficient information. Aalst et al. [70] explore this method and present a new algorithm to extract a process model from such workflow log (containing information about the workflow process as it is actually being executed) and represent it in terms of a Petri net. They also demonstrate that it is not possible to discover arbitrary workflow processes. discuss and analyse it with different real applications. VanDongen et al. [78] focus on the various Petri net based algorithms related to process discovery and compare them based on four different factors, which are supported control-flow constructs, assumptions about log completeness, supported levels of abstraction and underfitting/overfitting of discovered models. Five different approaches are presented by Aalst et al. [76] and explained by discussing different related problems. The first approach in this paper is based on Petri net theory using the α-algorithm which has been discussed previously. This algorithm has been proven to be effective for various types of processes in discovering the process workflow structure. There are also two tools which support this approach EMit and MiMo [76]. Wang et al. [81] proposed a different process mining algorithm called the λ-algorithm which mines events in logs that contain post-task information the information of post-task and the algorithm is based on an event multiset instead of an event trace. In this algorithm

the need to generate traces from the event log and to analyse them is eliminated since the event contains post-tasks information.

2.2.2.4 Process Mining in Healthcare Systems

The use of process mining in clinical environments is very challenging. There is not much research available that evaluates current algorithms in the real life clinical domain. Lang et al. [37] present an analysis of mining approaches with the capability of deriving process models from real working clinical data. For this purpose they examined the log files of the radiology information system consisting of CT scan, MRI, and X-ray of a clinic in Germany. After evaluating seven up-to- date algorithms (α-algorithm, α ++, heuristic, DWS, multiphase genetic and region based algorithms) they realised that most of them have problems when analysing event data from clinical workflows. Only four of them (α++, heuristic, DWS and genetic) could generate correct, but partially incomplete, models due to noisy and incomplete clinical log data. Thus, algorithms which rely on complete and noise free data logs failed to generate a model or a correct one. Maruster et al. [41], propose a method to automatically discover the workflow Petri nets from process logs which contain information about medical actions of a hospital. After experimenting with five different workflows, they found that the technique can be useful for parallel, conditional and sequential constructs but it cannot support all kinds of worflows such as cyclic or non-free-choice workflows. There has also been little research done on clinical pathway audit tools. Vanhaecht et al. [79] present a survey in which they select seven clinical pathway audit tools to determine if the clinical pathway audit tools can assess and identify clinical pathway documents. They concluded that the Integrated Care Pathway Appraisal Tool is the most useful audit tool in this regard.

2.3 Knowledge Representation

2.3.1 Context Representation

In order to be able to quickly identify relevant information, it becomes particularly important to take the context of the process participant into account. In fact knowing and utilising context information is a prerequisite to effectively provide relevant process information to process participants [43]. In recent years context models were the subject of several studies, particularly on context-aware computing, in which different approaches for context modelling were proposed [49]. Strang and Linnhoff-Popien [66] pointed out most of these approaches and classified the various models based on the data structures used to exchange and maintain contextual information in a given system. Sato [60] demonstrated a representation framework for contextual information critical for developing a methodological foundation for user-centred design practice. In the previous paper, the concept of context was demonstrated to be a critical resource for user-centred design practice. Najar et al. [49] review several context models proposed in different domains, content adaptation, service adaptation, information retrieval, etc. They propose a framework that analyses and compares different context models. This framework emphasizes the fact that the relevant information differs from one domain to another and depends on the effective use of the information.

2.3.2 Semantic Network

A well known technique of knowledge formalisation is the use of mathematical formulas [4], which are not easy to use and require an in-depth mathematical background [4]. Using semantic network as a tool for knowledge representation, is a more practical and simple approach [4]. With semantic network, knowledge is represented as concept nodes related by directional relationship links, like a graph [55]. This make exploring the framework of

knowledge as straightforward as moving from a node along one or more links to discover related information. The semantic network also makes extracting knowledge into human-readable format simpler. Discovering information about a specific topic in this structure is as easy as discovering relationships between two different objects [55].

There are six common kinds of semantic networks [65]: "(1) Definitional network emphasise the subtype or is-a relation between a concept type and a newly defined subtype. The resulting network, also called a generalisation or subsumption hierarchy, supports the rule of inheritance for copying properties defined for a super type to all of its subtypes. (2) Assertional network are designed to assert propositions. Unlike definitional networks, the information in an assertional network is assumed to be contingently true, unless it is explicitly marked with a modal operator. (3) Implicational network use implication as the primary relation for connecting nodes. They may be used to represent patterns of beliefs, causality, or inferences. (4) Executable network include some mechanism, such as marker passing or attached procedures, which can perform inferences, pass messages, or search for patterns and associations. (5) Learning network build or extend their representations by acquiring knowledge from examples. The new knowledge may change the old network by adding and deleting nodes and arcs or by modifying numerical values, called weights, associated with the nodes and arcs. (6) Hybrid network combine two or more of the previous techniques, either in a single network or in separate, but closely interacting networks."

Three characteristics of semantic network are described by Hartley and Barnden [27] as a) a way of thinking about knowledge that concepts are linked by relationships, b) a diagrammatic way of representing knowledge consisting of boxes, arrows and labels, c) a computer representation that allows database-like activity and a variety of inference techniques using algorithms that operate on the representations. According to Bessmertny [4] " The semantic network is an attempt to make the knowledge representation as close to the form in which it can supposedly be stored in human memory". It can be viewed as a relational graph consist-

ing of the subject-predicate-object. For example, learning about the patient history might be as easy as starting at the node representing the patient's name, moving along related links to find all the actions and examinations which the patient has gone through before visiting a specialist.

Semantic network have been used as a knowledge representation tool in different areas. Kulev at al. [36] present an algorithm for text classification using semantic network and describe a method for the extraction of relevant information from a given text. Semantic network was used by Niemann et al. [50] to solve the problem of image and speech comprehension. They present a framework to represent declarative and procedural knowledge with the goal of pattern interpretation. Much of the work behind semantic network is being done under Semantic Web project [3]. The project focuses on the problem of larger semantic network, the problem of large ontologies.

Semantic network is used as the basis idea of our research proposed model. It is selected as a simple tool for knowledge representation.

2.4 Provenance Management

Computing is identified as one of the main accelerator which has led to generating massive amount of data in different fields [18]. To analyse and understand the data, complex computational processes must be assembled which result in generating many final and intermediate data products. This adds more data to the overflow of data scientists need to deal with. For the purpose of data exploration, scientists and engineers need to put substantial effort managing data and recording provenance information to answer questions such as Who created this data product and when? When was it modified? What was the process used to create the data product?. Therefore, workflow systems grown and became very popular [18]. They support the automation of repetitive tasks as well as capture complex processes and capture process information for the derived data products. The provenance of a data contains

information about the process and data used to drive the data product. It provides documentation which is very important to preserving the data, to determining the data's quality to validate and reproduce the results. In the context of workflow systems, there are two forms of provenance which are prospective and retrospective [18]. Prospective provenance takes the specification of a computational task like a workflow, which are the steps that need to be followed to generate a data product. Prospective provenance, also known as workflow log, or process provenance [18], is metadata describing the workflows execution and associated service invocations; and data provenance, provides complementary metadata about the derivation history of the data product, including services used and input data sources transformed to generate it. Provenance allows scientists to monitor workow progress at runtime, thus it is very useful [20]. Reprospective provenance captures the steps that were executed as well as information about the execution environment used to derive a specific data product, which consists of a detailed log of the execution of a computational task [18]. According to Simmhan, Plale and Gannon [63] "Provenance is one kind of metadata which tracks the steps by which the data was derived and can provide significant value addition in such data intensive scenarios". Provenance (also referred to as lineage, pedigree, parentage) depending on where it is being utilised can be defined in various terms . Buneman et al. [7] explain data provenance in the context of database systems as the description of the origins of data and the process by which it arrived at the database. Provenance can be associated not just with data products, but also with the process(es) and services that enabled the creation of the data. Greenwood et al. [24] view it as data recording the process of experimental workflows, annotations, and notes about experiments. Simmhan et al [64] record uniform and usable provenance data that meets the domain needs while minimising the modication burden on the service authors and the performance overhead on the workow engine and the services.

The idea of process provenance is used in our proposed framework to assist representing

data generation as well as its evolution. In healthcare context, data provenance refer to evolution of patient history.

2.5 Semantic Annotation

Effect annotation helps improve the clarity and descriptive capability of the process model [23]. An effect is the result of an activity after execution. By referring to effect annotations, we can understand what knowledge we have after any of the tasks have been executed [23]. An annotated BPMN (which is introduced in section 2.2.1.2) model is one in which tasks and sub-processes have been annotated with a description of their immediate effect. Immediate effect can be accumulated throughout the process to provide a local in-context description of the cumulative effect at task nodes in the process [23]. For the purpose of obtaining functional effect annotations using informal annotation, Ghose and Koliadis [23] define a pair-wise effect accumulation process. Given an ordered pair of tasks with effect annotations, the cumulative effect can be calculated after both tasks have been executed in contiguous sequence [23]. Born et al. [5] present a tool that allows to annotate and automatically compose activities within business processes in web services context. Web services need to be formally annotated in order for tools to automatically compose them into an orchestration (defining the control flow between them). In a workflow execution engine, all tasks in the process have to be carried out manually or automatically by Web services. They aim to semantically annotate such tasks and to automatically discover or compose the services which collectively implement the required functionality. In this article they propose a way to solve the problem of composition within Web Services. A brief review of semantic annotation is provided, as it is applied in this research. It assist providing detail knowledge after any task is executed in CA-Nets.

2.6 Abductive Reasoning

In this sections a brief description is provided for three terms of reasoning, which are deductive, inductive and abductive. Then abductive reasoning will be discussed in more detail as it will used later to display the use of reasoning with CA-Nets.

In deduction reasoning [33], if something is true of a class of things, it is also true for all members of that class. For example, we have a true premise such as “all men are mortal”, then we have a true premise “Jack is a man”, then we conclude that “Jack is mortal”. Thus, based on the true premises we make a conclusion which is also true. The conclusion in deductive reasoning is certain and true. While in inductive reasoning the conclusion may not be true always. In inductive reasoning [22], many observations are made, a pattern is discerned, and finally a generalization is made which discerns an explanation or theory. In inductive reasoning general principles are derived from specific observations. For example, “Jack is a man”, “Jack is mortal”, then make the conclusion that “all men are mortal”. Abductive reasoning starts with an incomplete set of observations and proceeds to best possible explanaition for the observations. Abductive reasoning is explained below with more details as it is specifically chose in this study. This is because this study is hypothesizing the cause of an observed artefact which best explains that.

Abductive reasoning is used in many AI problems as a reasoning paradigm. The role of abduction has been demonstrated in various applications. It has been proposed as a reasoning paradigm in AI for planning, default reasoning and diagnosis [19]. Abduction, or inference to the best explanation, is a form of inference that goes from data describing something to a hypothesis that best explains the data [34]. This very broad definition covers a wide range of different phenomena involving some form of hypothetical reasoning. Abductive reasoning can be assumed to be a formalisation of hypothesis generating. Abduction within the realm of first order logic can be defined with the following schema:

Given a logical theory T [34] representing the expert knowledge and a formula O rep-

resenting an observation on the problem domain, abductive inference searches for a set of explanations and selects one of those explanations as E. For E to be an explanation of O according to T it should satisfies two conditions [19]:

O follows from *E* and *T*

E is consistent with *T*

In formal logic, *O* and *E* are sets of literals. The two conditions are defined as below:

$T \cup E \models O$

$T \cup E$ is consistent.

In other words, we can say: T is a collection of facts, E explains T such that no other E can explain T as well as E does. Therefore E is probably true [34]. In formal logic, O and E are assumed to be sets of literals.

Various frameworks have been proposed in abductive logic programming, given a knowledge base and some observations, providing possible sets of hypotheses that explain the observation. They use integrity constraints to generate the best and most consistent hypothesis. Integrity constraints are used for the purpose of hypothesis generation to make such sets consistent [1].

Diagnosis is often described as an abduction problem in AI [1]. The idea of diagnosis is to produce an explanation that best describe the patient's symptoms. More precisely, a diagnostic conclusion should be plausible enough to explain the symptoms and it should be significantly better than any other explanations. Based on the observation of symptoms, clinicians hypothesise possible alternative diseases which may have cause them [1]. To prove their hypothesis they prescribe some examinations or tests and pick those hypotheses confirmed by the test results [1]. We can then explain this kind of hypothetical reasoning process by explaining observation. Assuming causes of the observed effects and adapting such assumptions to subsequent events. For example, the discovery of new symptoms while treating patient.

In abductive logic programming, some formalism as well as proof procedures have been proposed to be able to generate, given a knowledge base and some observations, a possible hypothesis that explains the observations [19].

2.7 Process-oriented Information Logistics

Process-oriented Information Logistics (POIL) is proposed with the aim of providing the process-oriented and context-aware delivery of process-related information for knowledge-workers. Michelberger, Mutschler, and Reichert [44] proposed POIL to bridge the gap between business processes and process related information. POIL is consist of two different layers, integration and analysis. It is part of a bigger framework called niPRO (Personalized and Intelligent Process Portals). niPRO compromises four main layers:integration, analysis, navigation and visualization. In this study the focus is on POIL, because its the part which related to this approach. The structure of POIL is displayed in figure 2.1. It comprises two abstract layers: integration and analysis. The integration layer deals with the integration of data from different data sources, and analysis layer is responsible for creating semantic information network (SIN) [29].

The architecture of POIL is explained in [44] with more detail. The four architectural layers of POIL are defined as data layer, semantic integration layer, context layer, and application layer which are displayed in figure 2.2.

Details about each layer is provided in [44], but our main focus is on the semantic integration layer, which is responsible for the integration and analysis of both process information and business processes and is the most important core element of POIL. This layer integrates process information and business processes into SIN. A SIN is defined [29] as a labeled and weighted digraph $SIN = (V, E, L, W, f_l, f_w)$, where V is the set of vertices, E the set of edges, L the set of labels, W the set of weights, f_l the labeling function, and f_w the weighting function. The labeling function $f_l : E \to L$ assigns to each relation $e \in E(SIN)$ a

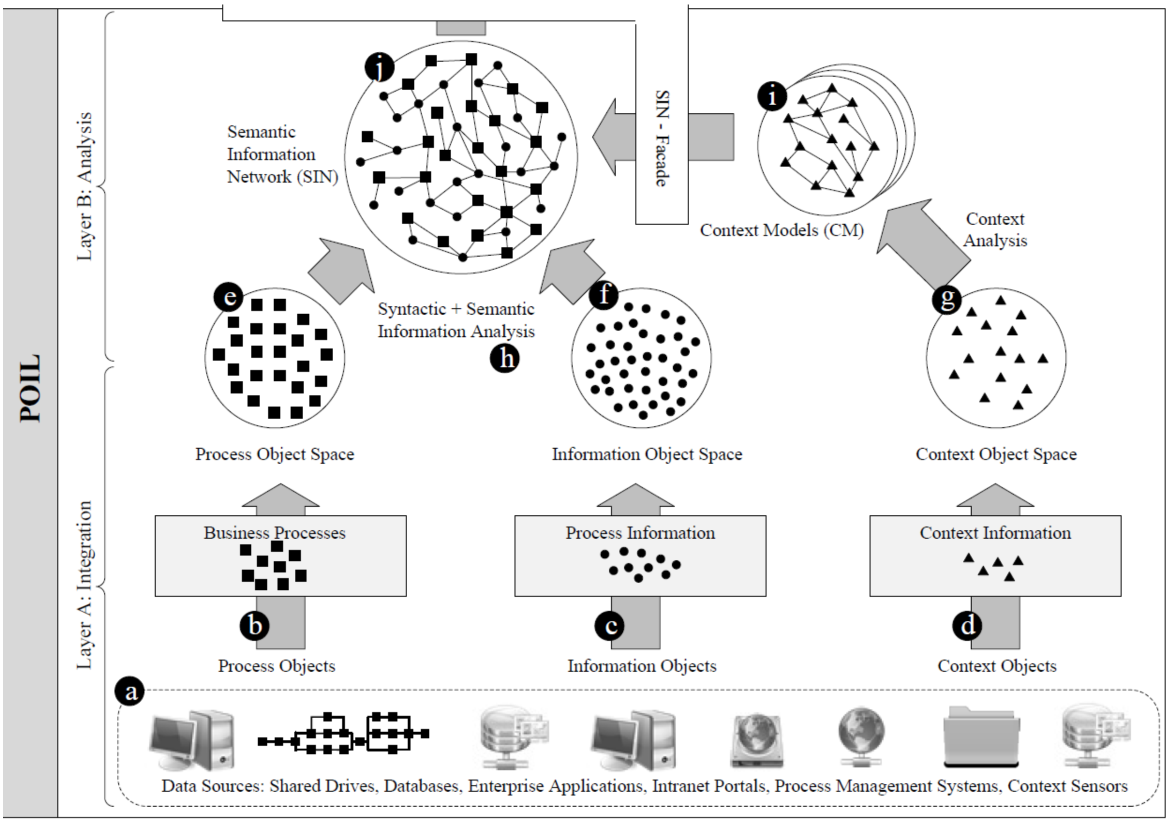

Figure 2.1: Process-oriented Information Logistics (POIL) (Source Hipp et al. [29])

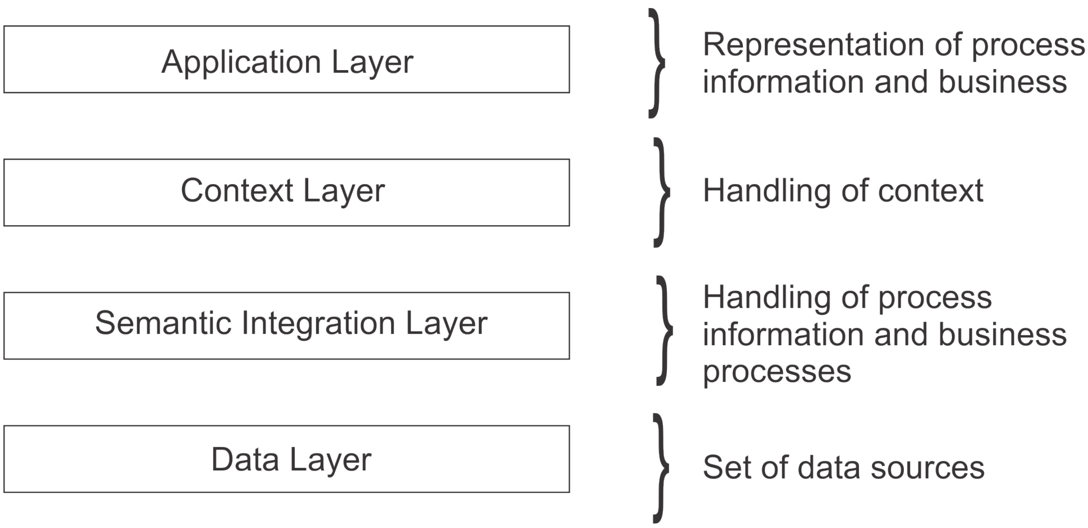

Figure 2.2: Interplay of POIL architecture levels

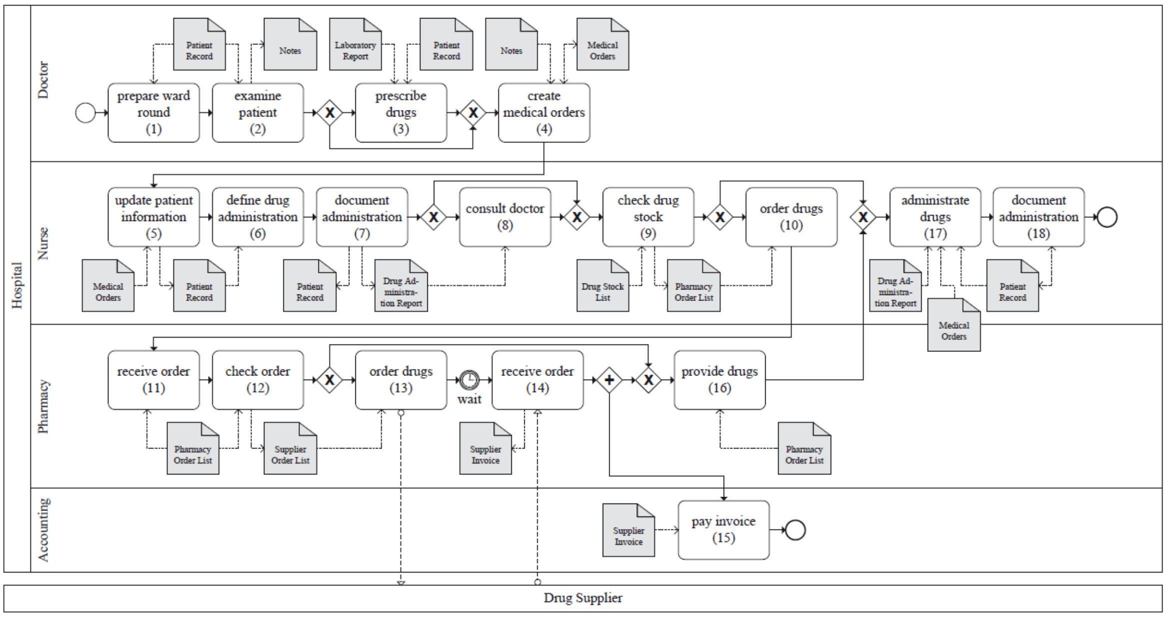

Figure 2.3: Procurement of drug, as described by Michelberg et al. [44]

label $f_l(e)$. In turn, the weighting function $f_w : E \rightarrow W$ assigns to each relation $e \in E(SIN)$ a weight $f_w(e) = [0, 1]$.

SIN is constructed based on a bottom-up approach and consists information objects, process objects, and relations between them [29]. Information objects are process information needed when working on business processes. Information objects include any medical reports, order forms, or patient records, and process objects include all relevant process objects such as tasks, data objects, sequence flows, gateways etc. Relations may exist between information objects, process objects and among information and process objects. Relations are labeled with the reason of the relationship and weighted with the relevance of the relationship.

Michelberger, Mutschler, and Reichert [44] explained the construction of SIN in six different phases in detail based on the case example shown in figure 2.3. We briefly discuss each phase in order to compare this approach with CA-Nets in chapter 4. In the first phase they specify business processes formally using a formal process modelling language such

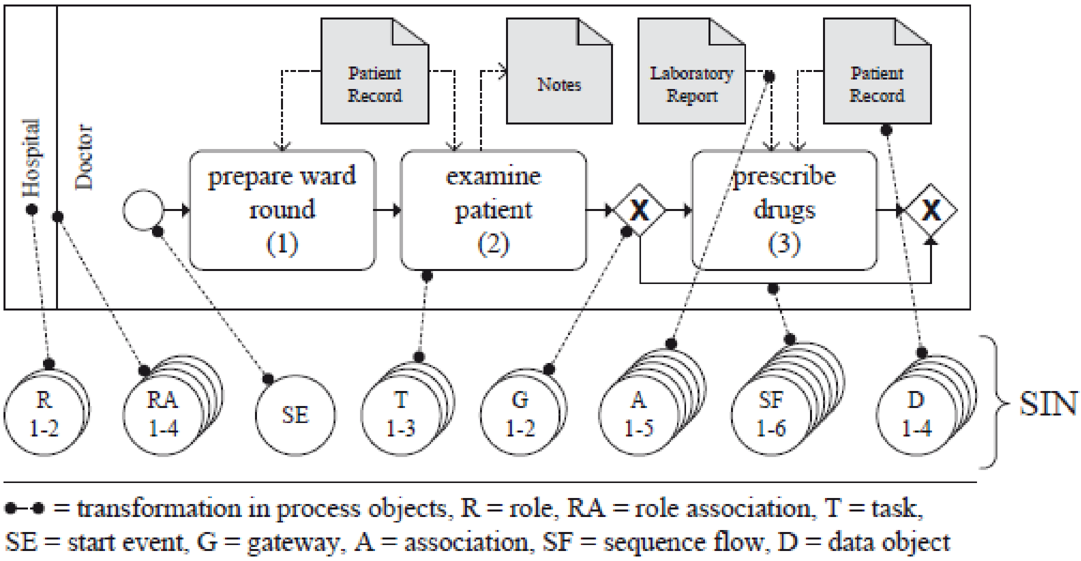

Figure 2.4: SIN phase 1, as described by Michelberg et al. [44]

as BPMN. They transform process instances into process objects (the algorithm for doing this procedure is not specified). They assume each object in the process model as an process object. The output of this phase is shown in figure 2.4.

In phase 2, the already existing SIN is extended by information objects. They assume that only process information from data sources that are connected by the data layer are integrated with the existing process objects. The output for this phase is displayed in figure 2.5.

Phase 3 and 4 deals with the identification of relationships between process objects and relationships between information objects, respectively. Any process objects of types sequence flow, association, role association are transformed into relationships in phase 3 where identifying relationships between process objects. Each relationship connects any two nodes and is labeled with a relationship reason and a relationship relevance. Algorithms such as text mining pattern matching and machine learning are applied. There is not enough information is provided regarding labeling the relationships.

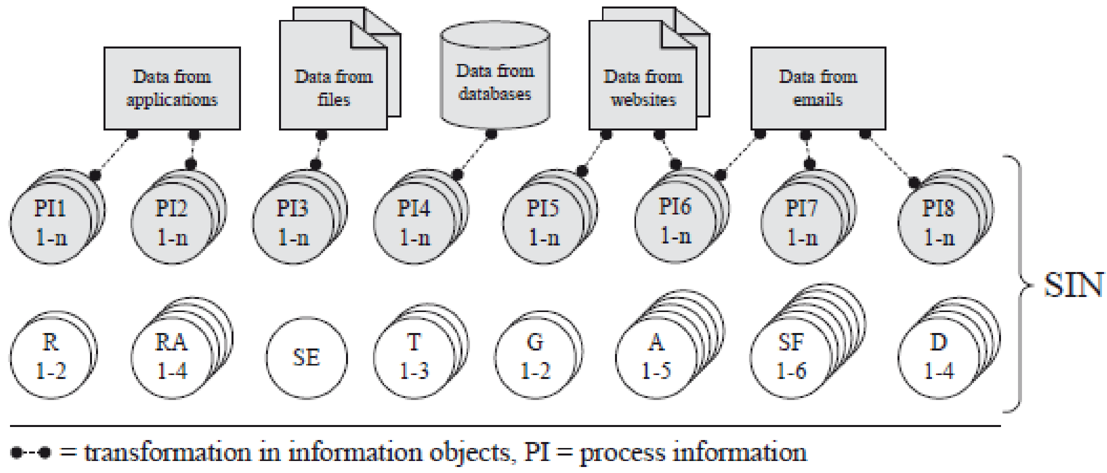

Figure 2.5: SIN phase 2, as described by Michelberg et al. [44]

Phase 5 deals with the analysis of relationships between process objects and information objects. In addition it detects further relationships considering metadata (e.g., author, keyword, content address). The last phase deals with the continuous determination of relationships among new and existing objects and validation checks of existing relationships. Figure 2.6 demonstrate two running process instances for the given process model in Figure 2.3.

As it is demonstrated in SIN, every business process objects of a process model such as tasks, data objects, gateways, etc. and corresponding process instances transform into a node in SIN.

In addition to the SIN, a context model is constructed based on available context objects which is discussed in detail in [44]. Context model is an ontology-based model and uses pre-defined context factors such as user, location, device or time. The context model enables representing all context information being relevant in the current situation of a process participant, which can then be used to filter the SIN [44]. The context model is completely independent from the SIN, and context objects are only stored in context model. Hence there exist a specific context model for each user, but a central SIN for all users [29].

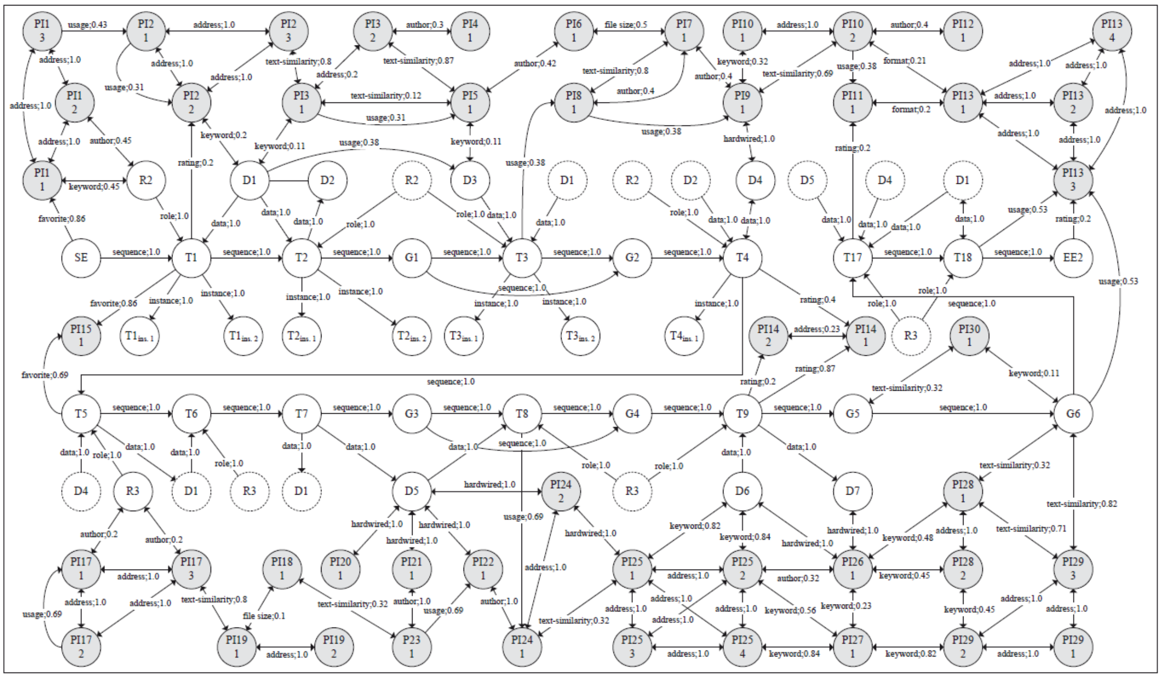

Figure 2.6: Final SIN,as described by Michelberg et al. [44]

Like SIN the context model is defined as a labeled and weighted digraph [29] $CM = (V, E, L, W, f_l, f_w)$, where V is the set of vertices, E the set of relations, L the set of labels, W the set of weights, f_l the labeling function, and f_w the weighting function. The labeling function $f_l : E \to L$ assigns to each relation $e \in E(CM)$ a label $f_l(e)$. In turn, the weighting function $f_w : E \to W$ assigns to each relation $e \in E(CM)$ a weight $f_w(e) = [0, 1]$ [29].

2.8 Conclusion

This chapter opens the debate relating to the importance of business process management in health care systems and the efforts to date which have been directed at solving the problems. Section 2.1 provides some background related to BPMN limitations and the need for tools for defining annotation to the processes. Section 2.2 describes the current circumstances of

process management in healthcare systems including an overview of it provide the existing issues. Section 2.3 reviews two of the main discussions in the knowledge representation field, which are context representation and semantic networks as a tool for knowledge representation. Section 2.4 provides information about provenance management. Semantic annotation is discussed briefly in sections 2.5.

Chapter 3

Clinical Artefact Networks (CA-Nets)

3.1 Introduction

This chapter describes the proposed framework which we have called Clinical Artefact Networks (CA-Nets). The resulting model will allow clinicians and other decision makers to navigate through the collection of data items that make up patient history just as one would navigate through a graph or a network.

The clinical processes supported by healthcare management systems are very complex and both produce and require large amounts of data [12], in the form of process descriptions, application data, forms, test results, images and reports, etc [44]. One of the major challenges today for healthcare systems is how to deal with the continuously increasing amount of disparate data [30, 44]. Such large quantities of unstructured data, spread across multiple data sources, makes it difficult and time consuming for clinicians to locate the information they need to perform their tasks efficiently and accurately. It is of prime importance not only in clinical decision making but also for the safety of the patients for clinicians to be able to retrieve the appropriate information, in the appropriate level of granularity, in a timely manner. The ability to retrieve this information is constrained by not only the sheer volume of data but also the logical disconnection between the physical storage of the data and the pro-

cesses that created the data as, in most situations, clinical processes and process related data are managed separately [44]. Healthcare management systems, shared drives and Intranet portals are used to organise and store the data, while clinical processes are designed and enacted by process management systems [44].

Clinicians and decision makers require information about a patient's previous treatments to assist in a range of clinical decision making. Moreover. different clinicians will require different context specific information at different times in a patient's journey through a hospital. Thus, in order for each clinician and decision maker to retrieve the appropriate information at the right time and at the right level of granularity, a context specific and dynamic data framework is necessary. For example, when an oncologist reviews a patients record, they are only interested in the information pertaining to the cancer disease they are treating and will not necessarily require details of other illnesses and other treatments that the patient has undergone in the past. Thus, an oncologist may ask questions such as: "Why has the patient undergone this particular test?", "Who performed the particular test (i.e. who created this data or performed this process?")", "Why was this test ordered (i.e. why was this process done?)" or, "What was the basis for a particular diagnosis?". To answer these type of queries the oncologist will need to go through all of the information about the patient distributed across various systems.

The other challenge in this context is to support different viewpoints of the various data consumers. As patients are referred to different departments of the hospital for different kind of services such as undergoing tests, or treatment plans, the patient history is reviewed by different clinicians with different purposes in mind. For example, there may exist number of different CT scan tests in patient's history but they could be related to different type of diseases. The tests may be recorded in a patient's record using the same synthetic name, but semantically will have different meanings and purposes in the patient history (this whole scenario is discussed in more detail in the motivation example in the following section).

Each test is likely to be reviewed by different clinicians for different purposes. Each clinician is interested finding reviewing information related to a specific treatment. For example, there are two CT scans in patient record. One of patient's heart, the other of patient's tongue. The oncologist needs to retrieve and view the scan of the patient's tongue (as the patient has been diagnosed with tongue cancer), while the cardiologist is interested in CT scan of the patient's heart. In order to be able to support different requirements of different clinicians, we aim to provide a dynamic and semantic framework to assist in retrieving data for different context specific queries.

CA-Nets utilise several technologies and techniques as its fundamental machinery, with a semantic network 2.3.2 at its core. A semantic network was chosen to be the basis of this framework as we found this to be one of the simplest ways of representing knowledge. Moreover, semantic networks could be used to represent patient specific knowledge in a related format to assist different kind of clinicians with different type of viewpoints. Data provenance [64] is used to provide complementary metadata prescribing the evolution of data sources and data items. Associated with data provenance are machineries to monitor and mine such data evolution, from the original data sources and why data items were created, to processes associated with the data and relationships between data items, through to the current representation [64]. In the healthcare context, data provenance captures the evolution of patient medical history. This technique is used in our research, to provide processes and data sources related to a particular patient health record that led to its creation and current representation.

3.2 Motivation Example

This section introduces a scenario from the clinical domain that will be used throughout the chapters that follow. This scenario is an extract from activities performed on patients in the local cancer service as the patients progress through the cancer healthcare pathway. The

scenario involves different processes (such as scanning patient history, patient examination and diagnosis) and comprises various process related information (such as patient records, test results, prescriptions, etc.) and decision making (such as the treatment plans for a patient). As a background, consider a patient's journey through a hospital, as the patient goes to different departments of a hospital to access necessary services. Each department has its own information system and the granularity of information recorded is different from department to department. Furthermore, it is quite normal to store patient's clinical data, but the sequence and the relationship between the processes and process related data is rarely recorded by healthcare management systems. For example, if a patient is prescribed to undergo a test, the test result may be saved in a different healthcare management system or in different files. Thus, the information available in the Electronic Patient Record (EPR) is insufficient for specialists to determine when or why a certain procedure has been performed. To obtain information about previously performed procedures (processes and process related information), specialists are required to refer to various data sources.

The scenario commences when a patient, suspected of having a cancer has gone through different tests ordered by a General Practitioner(GP) and is advised to consult an oncologist. Before any action takes place, the oncologist needs to review the patient history to be informed about the patient's status. For this purpose, the oncologist is given the patient health record which is usually a file consisting of all patient related data. The patient health record consists of the core data elements such as all prescriptions, patient symptoms, test results, images and reports, etc. Scanning all this data to find the relevant information requires a massive amount of time and effort on the part of an oncologist. For each document, such as a test result, the oncologist needs to search for its relationship to preceding and subsequent procedures and tasks. For example, when the oncologist looks at a CT-Scan result, he or she needs to know why the CT-Scan was prescribed. The sequence and the relationship of the tasks play a critical role in this scenario. Some critical questions need to be answered

like: “Why was this particular test ordered?”, “What was the temporal sequence of the test results?” Or, at “What stage of treatment is this patient?”. By answering these questions, the oncologist can better understand the patient’s condition and make decisions about the patient’s treatment type and schedule.

The scenario above, is described for a patient assuming that no other medical setback or complication occurs during a treatment or diagnostic process. Events like heart attacks, accidents like hand and foot injuries, reactions to a prescribed drug, etc. could occur which would require the patient to undergo an additional test(s) while being treated for another illness. This would require the patient to concurrently undergo two different diagnoses or treatment processes, which would make reviewing their patient history very complicated, thus making it more difficult for specialists or other decision makers to justify and find the reasons for the previously prescribed tests and treatments.

Similarly, when a patient dies during a surgery or any other clinical operation, before the review committee (who is responsible for finding the cause of death) starts their investigation, they need to know every procedure and task that has occurred and the reason for each task and treatment action. The sequence, the reason and the actor(s) associated with each task are each very critical in this context. For this reason, a comprehensive and continuous patient history should be provided to the user (in this case the review committee investigating a death) to help them answer such questions easily.

3.3 Clinical Artifact Nets

In this section, we discuss the CA-Nets framework. The resulting model will allow clinicians and other decision makers to navigate through the collection of data items that make up patient history just as one would navigate through a graph or network. Such a model will also be able to provide the appropriate information at the required level of granularity depending on the context. With this capability, the model is then in a position to answer

not only a range of queries based on patient data but also to provide the machinery to help clinicians in justifying clinical decisions.

A patient history is a set of patient specific data and it comprises all the processes, documents and information which was executed or generated during patient treatment. All the processes and related data exist in the various healthcare management systems, but they are kept separately. For example, patient prescriptions are saved and maintained in a file on system and any test results, whether images or documents are kept in a different file on the system and no link is defined within the system between a patient test result and the patient prescription. Thus, specialists need to spend a long time reading patient records and searching for the order that tests and activities were undertaken and the reasons behind the tests, and pre and post activities. The proposed CA-Nets provide patient history in a form that assists knowledge workers and decision makers to more easily identify the information they require based on their preference and current process context.

CA-Nets are defined based on semantic network technology [55] and consist of artefacts (nodes) and relationships (edges) between the artefacts. As stated in chapter2, in a semantic network, knowledge is represented as concept nodes related by directional relationship links, like a graph [55]. This makes exploring the framework of knowledge as straightforward as moving from a node along one or more links to discover related information. A semantic network also makes extracting knowledge into a human-readable format simpler. Discovering information about a specific topic in this structure is as easy as discovering a relationship between two different objects [55]. A semantic network is a way of representing knowledge by relating concepts with relationships [27]. In the case of CA-Nets, we represent clinical knowledge by artefacts (clinical data and processes) instead of concepts and the relationships between those artefacts.Thus, we propose to use a semantic network technique to establish our framework (CA-Nets) to represent the clinical knowledge for a set of generated artefacts based on clinical data and processes. Relationships between these

artefacts are defined to support different types of queries on the patient history in a clinical context.

If we were to take a traditional annotated business process model (bpm), then every object in the bpm will translate to a node in a graph. In CA-Nets every document (such as a blood test result, CTScan, prescription, MRI image, etc.) and every process (such as diagnosis, prescribed treatment plan, tests, consultation, etc.) is identified as an artefact. These artefacts are related to each other via a set of relationships. The value of each relationship specifies the type of relationship that exists between any pair of artefacts. The relationship types, which are based on the context, consist of: basis-for, because-of, contribute-to, update, etc. The resulting network makes exploring the framework of knowledge as straightforward as moving from an artefact (clinical data item or process) along one or more links to discover related information [55]. Thus, the resulting network makes the job of finding related information and relationships between disparate information easier. Such networks represent information and link information in a format which enables analysis and context aware selection of required information. A specific process context provides the information related to that specific node in the network and each node can be a starting point for the process analysis.

The idea of CA-Nets, as an intermediate layer between processes, process related information and querying knowledge is represented in figure 3.1. The top box shows the distribution of information for all the patients across different systems. For example, patient prescriptions are saved and maintained in a file on system and any test results, whether images or documents are kept in a database and the patient next appointment is saved on a different healthcare management system. The second box, CA-Nets, displays knowledge for a particular patient in a semantic format. This framework is a layer between the information in the system and the different queries for which the clinicians require answers from the patient history. CA-Nets are constructed on the fly, and dynamically represent the information

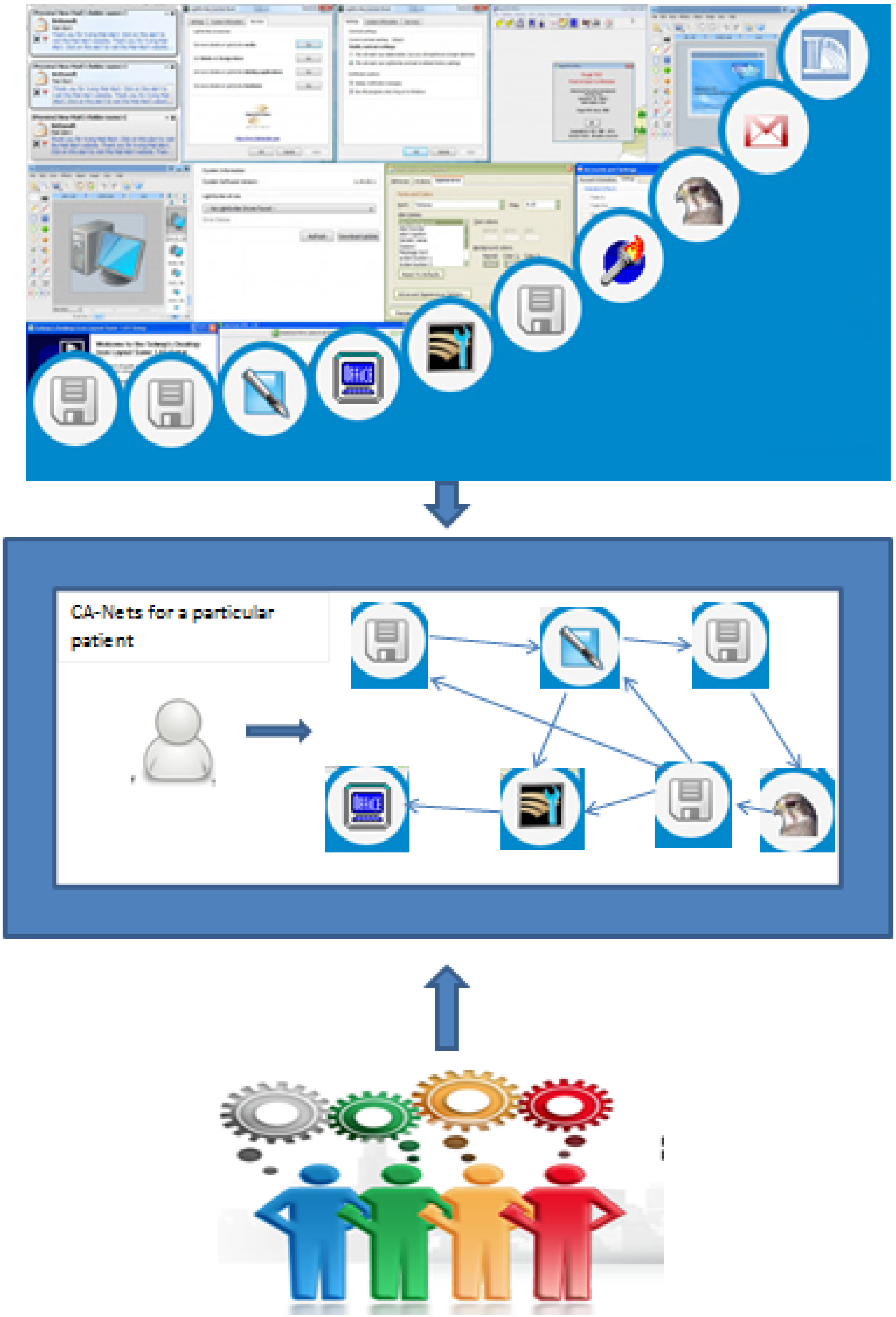

Figure 3.1: CA-Net layer

about a particular patient. CA-Nets correlate the semantic aspect of the data irrespective of how, where and in what format the data are stored. The resulting model will then provide an ability to navigate through the collection of data items as one would navigate through a graph or network. With this capability, the model is then in a position to answer not only a range of queries but also providing the machinery to help justify clinical decisions.

A formal definition of multigraphs is provided below as a basis for the formal definition of CA-Nets. The concept is discussed in detail with an example.

Definition 3.3.1. [83]**[Multigraphs]** A multigraph is defined as M = $\langle N, E, f \rangle$

- N is a set of nodes
- E is a set of edges
- $f : E \rightarrow \{\langle u, v \rangle : u, v \in N \text{ and } u \neq v\}$

If $e_1, e_2 \in E$ are such that $f(e_1) = f(e_2)$, then we say e_1 and e_2 are multiple or parallel edges.

A multigraph is a graph which is permitted to have multiple edges with the same end nodes. A graph is a representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are called nodes, and the links that connect some pairs of nodes are called edges. It allows for multiple edges between a pair of nodes, but the edges do not allow for loops,that is, an edge that connects a node to itself.

A CA-Net is a directed multigraph with a set of artefats(A) instead of set of nodes(N) and a set of relationships (R) instead of a set of edges(E). In our study, artefacts can be any type of process, or process related information. The relationships are edges between the artefacts which specify relationship reasons. Function f, which identifies the edges between the nodes in the multigraph, is reflected in CA-Nets with the according function f.

Definition 3.3.2. [CA-Net] A CA-Net is a labelled directed multi-graph M = $\langle A, R, f \rangle$ augmented with nodes and edge types and is defined as an 8-tuple $\langle A, R, f, T_A, T_R, l_a, l_r \rangle$ where:

- A is a set of nodes representing the set of artefacts
- R is a set of edges representing the set of relationships
- $f : E \rightarrow \{\langle a,b \rangle : a,b \in A$ and $a \neq b\}$
- T_A is a set of artefact types
- T_R is a set of relationship types
- $l_a : A \rightarrow T_A$ maps every artefact to its type
- $l_r : R \rightarrow T_R$ maps every relationship to its type

The definition above permits a pair of artefacts in a CA-Net to be related via multiple relationships of the same type. This may be undesirable in general. We therefore propose an additional condition to obtain a *well-formed* CA-Net.

Definition 3.3.3. [Well-formed CA-Net] A CA-Net $\langle A,R,f,T_A,T_R,l_a,l_r \rangle$ is said to be **well-formed iff** for each pair of relationships $r_1, r_2 \in R$:

- $f(r_1) = f(r_2)$
- $l_r(r_1) \neq l_r(r_2)$

Example 3.3.1. The CA-Net for a particular patient's journey through the hospital is demonstrated in figure 3.2. For clarification, we demonstrate the process artefacts in red and process related information artefact in blue. This distinguishes between processes and process related artefacts more easily. The CA-Net demonstrates knowledge about a patient who had previously been diagnosed with colon cancer. The patient is referred to the doctor and presents with some symptoms which prompt the doctor to request a CT scan. The corresponding artefacts can be seen in figure 3.2. It starts with the patient condition artefact which shows that the patient was previously diagnosed with colon cancer. The patient condition artefact is followed by the current patient symptoms. The patient's symptoms lead to

a CT scan test. Then, based on the CT scan result (which is generated as result of CT scan test), the patient was diagnosed with tongue cancer. The process diagnosis results generate an artefact following the test result artefact. Based on the diagnosis, the specialist prescribes an appropriate treatment plan. The patient condition is also updated each time and accordingly a new artefact is created. After each treatment process, a treatment report is generated which results in the creation of a new artefact of the type treatment report in the CA-Net. After each referral of the patient, the patient condition is updated as illustrated in figure 3.2. Each time clinicians require information about a patient, they will need to find the related artefact and follow the relationships in order to find the answers to their questions. The elements of the set of artefact types for this example include:

- Patient condition: processes and documents related to patient condition which specify current and previous patient status
- Patient profile: data related to patient identification and contact information
- Symptoms: data related to patient symptoms
- Test result: documents which are produced as result of a patient undergoing a test, such as: CT scan, blood test, MRI, etc.
- Performdiagnosis: processes related to the diagnosis procedure
- Treatment plan: set of data related to patient treatment
- Treatment process: set of tasks related to patient treatment plan
- Treatment report: prescriptions or any data regarding patient treatment plan

The elements of the set of relationship types for this example include:

- contributes-to: This relationship exists when one artefact is related to other artefact as a basis or to complete the available knowledge.

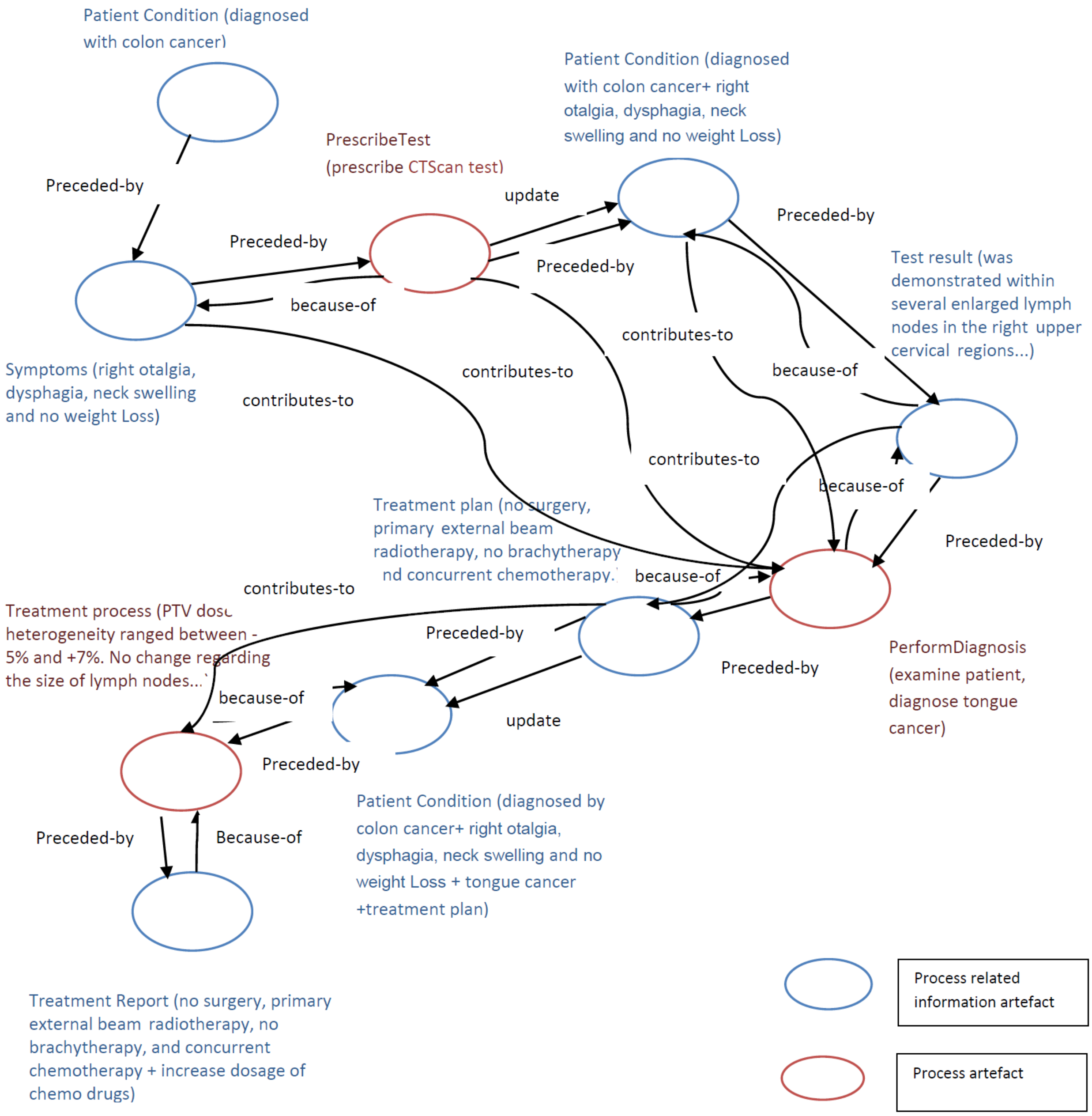

Figure 3.2: CA-Nets for a particular patient

- preceded-by: This is used to describe temporal relationships - in particular, that an artefact was executed or generated at some point before another artefact.
- because-of: This type of relationship exist when one artefact is performed because of

the result of the other artefact.

- update: This relationship exists when one artefact result should update another artefact state.

The above artefact types are identified based on considering different possible scenarios which could happen for a particular patient. For this purpose, we considered possible scenarios that would prompt a patient to consult a doctor. In each scenario we identified the major possible types of performed processes and generated documents related to a patient. Then we identified the set of artefact types based on the different processes and process related information. For example, we assumed a patient would consult a GP in the case of a heart attack. First, the GP would read the patient profile and review the patient's condition and examine the patient's symptoms and prescribe a CT scan test. This scenario resulted in identification of major processes and process related information such as: process profile, patient condition, symptoms, prescribing test and, consequently, a CT scan test result. The list of aforementioned artefact types are the major identified processes and process related information based on different scenarios in a healthcare system context.

Likewise, to identify possible relationship types, we considered how and why artefact types could be related and proposed terms which to describe the reasons for a relationship between any two types of artefact. Moreover, we considered the types of different queries that would need to be supported. In this way, we identified the five major types of relationships listed above. These lists could grow further if become more richer if we needed to consider supporting other different types of queries.

Example 3.3.2. In this example, we discuss example 3.3.1 considering possible common concurrent treatments for a patient as discussed in the motivation example. In this example, we illustrate a CA-Net for a complex situations whereby tracing patient history and justifying diagnoses and treatments is complicated. We applied the case of a possible concurrent illness or accident that may occur for any patient while in the process of being treated for

another illness in figure 3.3. In this example, after the patient had undergone a CT scan of

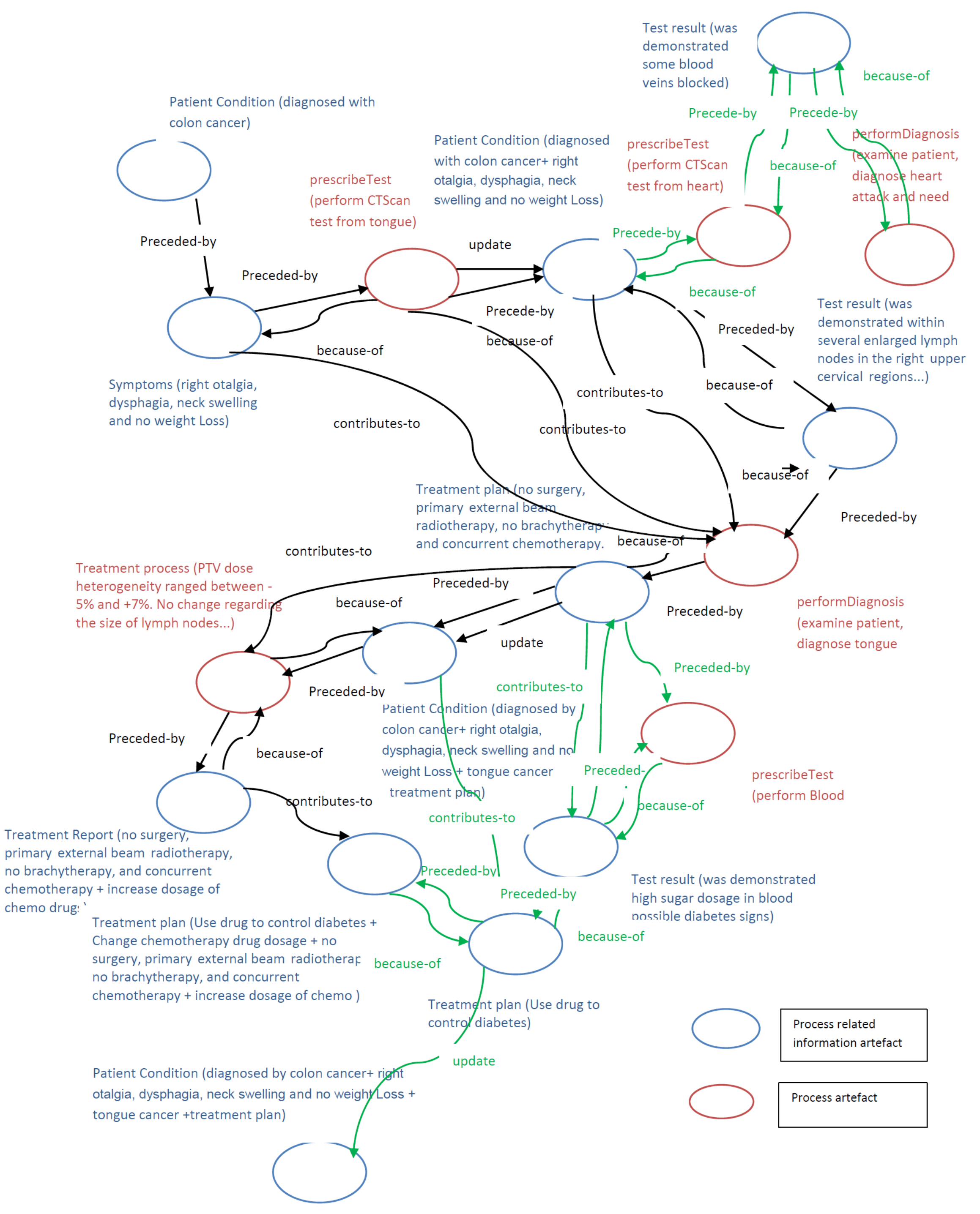

Figure 3.3: CA-Net for a particular patient

his tongue, another CT scan was performed on his heart. Synthetically, the two tests are the same, but they have different semantic meanings. The first test is prescribed for the patient, as a result of symptoms such as right otalgia, dysphagia, neck swelling and no weight loss. The second test is prescribed after the patient has had a sudden heart attack. The result of the heart CT scan shows some blockage in his veins. In figure 3.3 we demonstrated the second concurrent illness connected with relationships shown in green to distinguish that they are for different purposes. Moreover, the relationships in green illustrate how unforeseen problems (that are possibly unrelated to the original condition) can make the job of reviewing a patient history much more difficult. All the concurrent and perhaps some irrelevant processes and documents are recorded, but finding the reasons associated and following the related processes and documents is not an easy job.

In this example while the patient was undergoing treatment, a change in the patient's condition resulted in the doctor prescribing that the patient undergo a blood test. The blood test result showed that a drug the patient was using had resulted in an increase in the patient's blood sugar. Thus, a new treatment plan was prescribed for the patient which was appropriate for both illnesses. This example illustrates how CA-Nets can assist a doctor in reviewing a patient's history, by representing the sequence of and the reasons behind previous processes performed.

3.4 The motivation for using CA-Nets

In this section, we discuss in detail the motivations for using CA-Nets. They can be used to capture and represent knowledge in a contextual format in a healthcare system. They support a range of viewpoints from a general perspective such as patient's journey through the healthcare system to more specific perspectives like monitoring specific patient treatment protocols. CA-Nets can help a specialist to be precisely informed about the sequence, and relationships associated with previous tasks performed by other medical professionals.

Moreover, they can assist in decision justification and provide a rich basis for data mining. Patient treatment monitoring is something else that CA-Nets can provide information about. Each of the aforementioned motivation items are discussed below in detail. They are explained using examples 3.3.1 and 3.3.2.

3.4.1 Obtain Patient Context

The most important and obvious feature of CA-Nets is to make the act of obtaining or extracting context specific patient information easy. By representing the related information in a semantic network, users retrieve the relevant information, quickly and easily and can make decisions based on the information retrieved. For example in figure 3.2, when the specialist needs to know why the patient was diagnosed with tongue cancer, he or she can follow the relationships from the diagnosis artefact. Moreover, in figure 3.3, if the specialist needs to know why the treatment plan for tongue cancer was changed, then he or she can follow the relationships from the treatment plan to discover the reason. By tracing the relationships, the specialist would find the blood test result which shows the increase in blood sugar that prompted and as a result a new treatment to be prescribed.

3.4.2 Decision Justification

In the health domain, almost all treatment of patients is based on the decision of doctors or specialists. It is common for mistakes to occur as result of limited knowledge, being under pressure or negligence. So there is a need for an approach to enable monitoring and evaluation of a doctor's diagnosis and treatment actions. For example, an evidence based approach which could provide answers to questions like: "Why did this error happen?", "What was the cause of this mistake?" Or, "Why did the doctor made that decision?", etc. Based on decision justification theory [14], which says: faced with a poor decision outcome, individuals ask themselves whether the process or decision that led up to it was justified or

not. If the answer is positive, based on the seriousness of the outcome they feel regret. In figure 3.2, if a specialist first wants to review and discover the reasons behind why the patient was diagnosed with a tongue cancer, she or he needs to find the diagnosis artefact and follow the relationships to the previous artefacts. Then, if the specialist wants to know the justification behind why the patient was prescribed to undergo the CT scan, she or he can easily follow the relationships from the CT scan test artefact to the previous artefact.

In figure 3.3, when the specialist needs to know why the patient has undergone two CT scans, by following the relationships from the CT scan tests, the specialist can find out about the sudden heart attack which resulted in the second test being ordered.

3.4.3 Support the Extraction of Specific Viewpoints

As discussed earlier in the motivation example, when a patient with a diagnosed disease visits a specialist, what a specialist needs to read about is any information related to that specific disease such as any related information which may clarify why the patient has been diagnosed with the particular disease. For example, if the doctor seeks the history of a patient's cancer related data such as MRIs, or blood tests, he doesn't need to know about a common cold she or he had last month. CA-Nets enable the doctor to query and find that specific information directly without wasting time on unwanted or unrelated information. Moreover, different clinicians such as doctors, nurses, radiologists, oncologists, etc. may review a patient's history but each of them will require different types of information related to their specific task, role and skills. Thus, each of them will review the patient history with specific viewpoints. For example, in example 3.3.2, after the patient undergone a CT scan related to tongue cancer, he or she underwent another CT scan test as a result of a sudden hearth attack. When a heart specialist reviews the patient history, she or he is more interested in the second CT scan test, which is a heart CT scan. Whereas an oncologist needs to know the information related to the tongue cancer and therefore the CT scan of the

patient's tongue.

3.4.4 Provides a Rich Basis for Data Mining

Process mining attempts to provide a concise assessment of reality which can be helpful in verifying process models and process redesign efforts. CA-Nets are not only useful for knowledge representation, but also for the storage and retrieval of knowledge. In real situations, as each process is executed, it is stored in the process log and can be retrieved using process mining techniques (which is discussed in detail in section 2.2.2.3). By reading process logs which are provided by process modelling engines, and representing the required knowledge, CA-Nets provide a rich basis for data mining. Thus, CA-Nets can precisely represent the sequence of the real processes which are performed for a particular patient.

3.4.5 Treatment Monitoring

Monitoring and tracking patient treatment could give us the confidence that a patient is safe and on the right track. By tracing and analysing the artefacts in the CA-Nets, clinicians can verify the patient treatment process. For example, in figure 3.2 the specialist can follow the relationships from the diagnosis artefact to the treatment plan to find out if the prescribed treatment was appropriate for the patient. Moreover, if a specialist need to know about the treatment processes, this can be achieved by tracing the artefacts forward. In figure 3.3, if the specialist wants to monitor the treatment plan and discovers the change in the plan, he/ she can trace the relationships from the new treatment plan.

3.5 Conclusion

This chapter proposed a framework to represent knowledge for a patient specific treatment to support different type of queries. CA-Nets are described based on the semantic network

technique and the idea of data provenance. A major goal of CA-Nets is to introduce an approach to help in the management and retrieval of related data in healthcare management systems. CA-Nets provide information about a patient's current and previous treatment in a specific context that assists clinicians in a range of clinical decision making. The motivating example was provided based on research and analysis of the activities performed on patients in the local cancer service. This example was used as a basis for the motivation behind the idea of CA-Nets, and will be used through out the entire thesis. A formal definition for CA-Nets was provided and the concept illustrated with examples. Ultimately, this chapter discussed the motivations for developing CA-Nets and showed the potential benefitis, to clinicians and therefore also to patients, of implementing CA-Nets in a healthcare context.

Chapter 4

Representing Knowledge and Reasoning with CA-Nets

4.1 Introduction

In chapter 3, an approach to represent knowledge about the patient-specific treatment context in a manner that supports a range of clinical queries was presented. This chapter provides an explanation of how CA-Nets are created from readily available information about patient-specific treatment. First, the major elements that make up a CA-Net, such as process log and knowledge base, are defined. Second, the generation of CA-Nets is discussed. Then the types of queries which CA-Nets can support and answer are described, followed by a discussion about supporting reasoning in CA-Nets via the use of effect annotation. Finally, a comparison between the approach presented in this thesis and another related approach is provided.

4.2 How to create a CA-Net

The creation of a CA-Net depends on the underlying business processes. The business processes are represented using a formal process modelling language. In this study it is assumed that business processes are represented via the industry-standard Business Process Modelling Notation (BPMN). In a real situation, as each process is executed within a business process engine, is the business process will be stored in the process log and can be retrieved using process mining techniques, (which is discussed 2.2.2.3 in detail). Processes can be retrieved by reading the process log which is provided by the process modelling engines. CA-Nets precisely represent the sequence of the processes which are performed for a particular patient. When retrieving the sequence of the executed processes, each process

Table 4.1: Process Log

Number	Process Id.	Time	Performer
1	P1	T1	Per1
2	P2	T2	Per2
3	P3	T3	Per3
.	..	..	..

creates an artefact in a CA-Net.

The process log (Table 4.1) consists of data including the order, timestamp and performer of each event executed. The data in the process log is used to assist discovering the sequence or concurrency of the executed processes. Table 4.1, presents the log table identification number (Number), the process identifier(Process Id), the timestamp (Time) showing the time at which the process commenced execution and who performed the process (performer).

In addition to process log, the existence of a background knowledge base (KB) that provides a basis for artefact types and artefact relationship types is assumed. The information in the knowledge base consists of a set of rules. The rules provide different types of information such as the type of artefact and the types of relationships, and thus apply the background knowledge required to specify the artefact types when they are generated in CA-Nets. For

example, when a process such as prescribing a test for a patient is executed, a new artefact with the artefact type 'Prescribe a test' is generated. Or, if a doctor examines the patient, it results in generation of an artefact with the type 'patient examine'. In addition, the knowledge base rules identify the correlated process information that should be generated before or after any artefact. The rules specify that if an artefact of type α is generated, then an artefact of type β is expected. For example, if the patient undergoes a test, we expect to have another artefact with the type 'test result' after-wards. Thus, these rules can help to pinpoint missing artefacts in a CA-Net. The rules in the knowledge base are incomplete will evolve as more CA-Nets are generated and more artefact instances are observed which in turn will help to generate more rules. (For more information please refer to section 4.3.)

Based on the type of any two given artefacts, the timestamp of the generated artefacts and the type of queries supported, the relationship types are specified. When a new artefact is generated in a CA-Net, the types of relationships that relate the new artefact to other artefacts must be identified. Four major categories for relationship types are identified according to the examples that are used to explain CA-Nets in this study. An artefact can be generated as a result of another artefact, preceded another artefact, update another artefact, or contribute in some way to another artefact. The four categories of relationship types are defined below:

- Temporal relationships: are relationships between two consecutive artefacts based on the time stamp of the artefact generation. This type of relationship is identified between any new artefacts and the most recent existing artefact in the CA-Net.

- Causal Relationship: are relationships that exist when an artefact is generated as a result of another artefact, according to the rules in the knowledge base. For example, the rules in knowledge base specify that a test result artefact is generated because of a Prescribe-test artefact. The relationship type "because-of" is of this class of relationships.

- Operational Relationships: are relationships that exist when a process or set of pro-

cesses performed result in a new artefact or update to an existing artefact. The relationship type "update" is of this class of relationship. This type of relationship relates any artefact with the artefact type patient condition. It means that after a set of processes is performed, the patient condition profile, which contains patient health information, is updated.

- Contribution Relationship: are relationships that exists when one artefact is related to another artefact to complete the available knowledge or contributes to the generation of the other artefact. Therefore, this type of relationship is generated when a new artefact is related to an existing artefact but is not generated consecutively directly after the other artefact.

The other factor which is important when considering relationship types between two artefact types is the type of queries which must be supported. Queries such as "Why was this test prescribed?" or "What was the basis-for this diagnosis?" or "What happened after this test result?" are major types of queries clinicians may ask when reviewing a patient history. For example, when a doctor finds a PrescribeTest artefact he or she needs to know why this test was prescribed, so he or she will start at the PrescribeTest artefact and look for a causal relationship which could answer hos or her query.

An explanation of how to create a CA-Net will be based on the following example.The various processes that describe a patient being referred to a doctor at an abstract level are represented in figure 4.1. Each time the patient is referred to a doctor, the doctor reviews the patient history, then examines the patient and documents the patient's symptoms. Then, based on their observations of the patient's condition, the doctor may prescribe a test for the patient, or if the patient has been referred to the doctor to find about their test results, then the doctor will review the patient's test result and may prescribe some drugs or treatment for the patient. The process of prescribing test and checking test results may be repeated several times in order for the doctor to make a diagnosis.

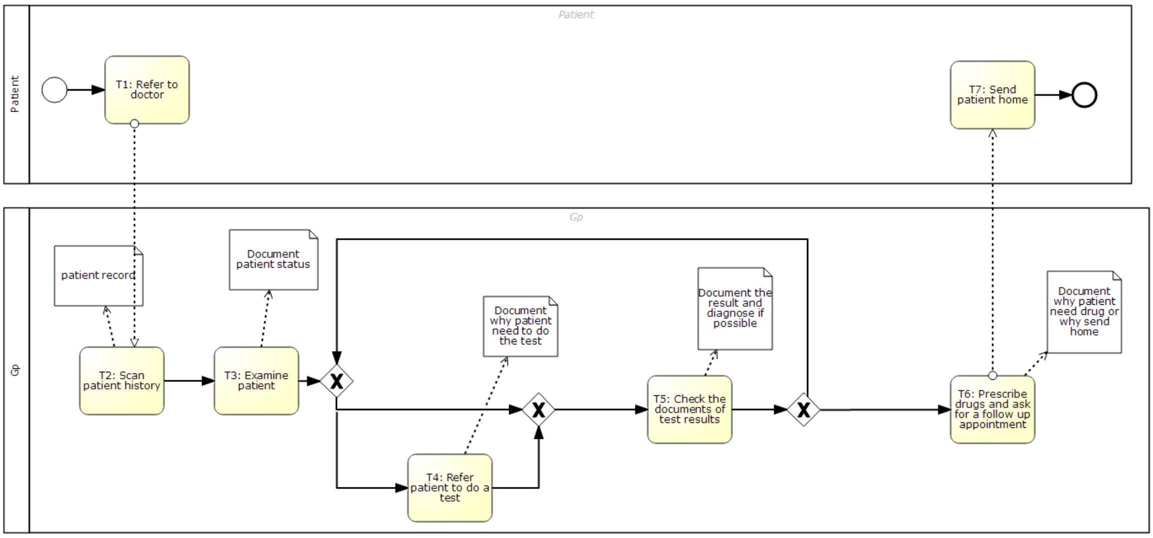

Figure 4.1: Clinical Processes for a patient

Upon execution of any process the corresponding artefacts are generated in a CA-Net. Every time the patient is referred to a doctor, the required processes and process related information are generated. As any new process is executed or any process related information is generated, subsequently a new artefact(s) is created in the CA-Net.

Once a new artefact is created, then the set of relationship types that exist between the new artefact and any existing artefact must be identified. In other words, the following question must be answered: is there a relationship, from the known repertoire of relationships, that can connect these two artefacts? For this purpose, for any two artefacts in the set of artefact type A, we examine every relationship type that exists in T_R. For any pair of artefacts with specified artefact type, timestamp relationship types are identified according to the type of queries that exist between the two artefacts. For example, in figure 4.2, the patient presents with some symptoms that prompted the doctor to request a CT scan. This process resulted in the creation of new artefacts with the types ‘symptoms’ and ‘perform test’. Then the types of relationship between the two new artefacts in the CA-Net were identified. In this example, two relationship types were identified between the artefact perform test and

the artefact symptoms which are preceded-by and because-of. The relationship preceded-by is identified as artefact 'perform test' is generated after the artefact 'symptoms', and because-of is identified because it provides answer for queries such as, "Why was this test prescribed?", and " What was the basis for this test" and " What action is performed after the symptoms are identified?". The elements of the set of relationship types for these examples, and how the relationships are identified for any pair of artefacts, are explained below.

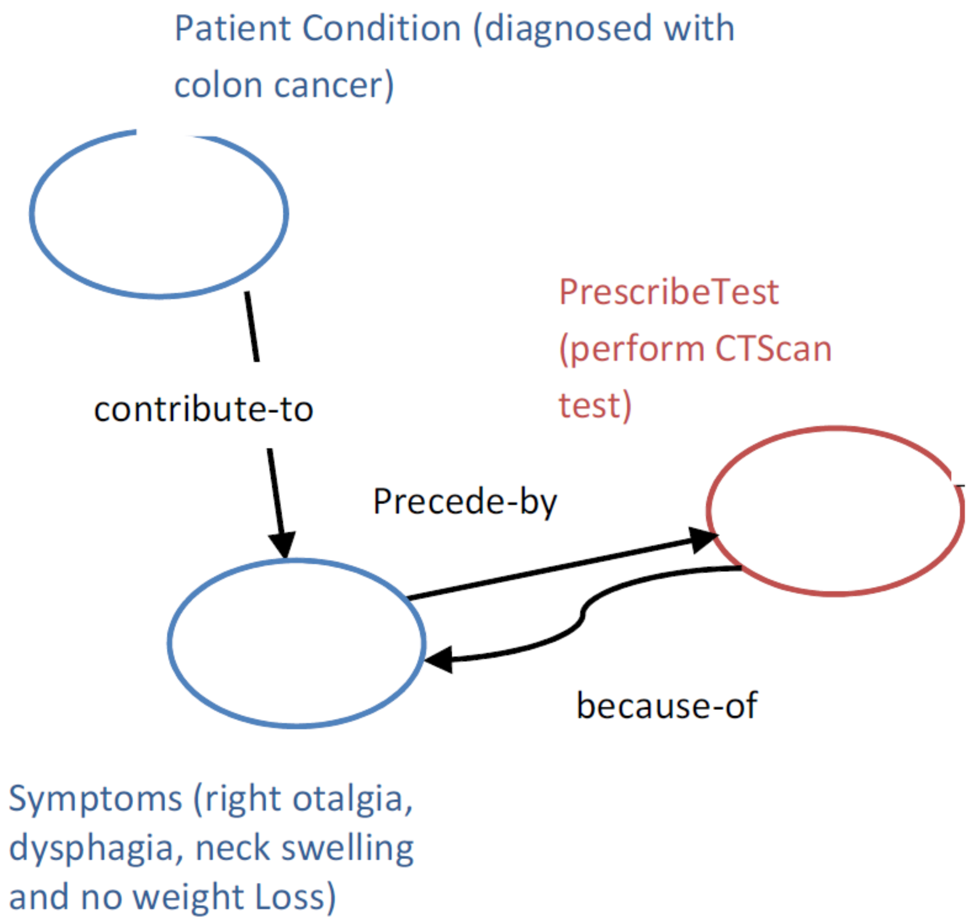

Figure 4.2: Relationship types in a CA-Net

- preceded-by: This relationship type is used to describe temporal relationships, in par-

ticular, that an artefact was executed or generated at some point before another artefact.

This relationship type relates any new artefact and the most recently created existing artefact in a CA-Net. Any two consecutive artefacts in a CA-Net can be related via this type of relationship. This specifies that when adding artefact type $performTest$ to a CA-Net we relate the previously created artefact which is $Symptoms$, to the new artefact (figure 4.2). This indicates that the artefact $Symptoms$ is preceded-by the artefact $performTest$. The last artefact in a CA-Net is identified according to the time stamp in the process log. Based on the time of creation of each process or process related information in the process log, we generate the corresponding artefact in the CA-Net.

This type of relationship is identified between any two consecutive artefacts whereby the second artefact is executed or generated after the first artefact. Moreover, if any other types of relationship exist between the two artefacts, we also put the other type (if any exist) between the two artefacts. Multiple relationships between the two artefact are allowed as CA-Nets were defined as multigraphs earlier in chapter 3. For example, in figure 4.3, two relationships were identified between $< prescribeTest,$ $patientCondition >$. One of them is of relationship type update(which is described later in this section) and the other relationship type is preceded-by. It means that $patientCondition$ is updated after the $prescribeTest$ process artefact is created. In figure 4.3, we can follow this type of relationship between any two consecutive artefacts. As another example, the $Testresult$ artefact is preceded-by the artefact type performDiagnosis. This shows that after observing a $TestResult$ the doctor $performedDiagnosis$. After the doctor performs a diagnosis, then he or she will make a $TreatmentPlan$, thus resulting in a $Preceded-by$ relationship type between $< PerformDiagnosis,$ $TreatmentPlan >$. The patient condition is subsequently updated, and generates a

$< Preceded - by >$ relationship between $< TreatmentPlan, PatientCondition >$. The patient condition is updated and a treatment process is performed, which means a $Precede - by$ relationship exist between $< PatientCondition\ , TreatmentProcess >$.

With this type of relationship, we can support queries such as “What is preceded-by this performTest?” or “What was the basis-for this test ?” or “What was the nearest previous performTest for this testResult artefact?”.

For every artefact type in our example, we now identify whether relationship type $< preceded - by >$ can relate this artefact to other artefacts as this type of relationship only exists between consecutive artefacts which are identified based on the artefact time stamp.

- Patient condition
- Symptoms
- Test result
- performDiagnosis
- prescribeTest
- Treatment plan
- Treatment process
- Treatment report

- because-of: This type of relationship exists when one artefact is performed as a result of the other artefact. In figure 4.2, the artefact $performTest$ is executed as a result of the patient symptoms. The doctor prescribes a test because of the patient’s observed symptoms. This type of relationship relates artefacts which trigger creation of another artefact or basis for the other related artefacts. In figure 4.3, when the doctor observes the patinet’s symptoms, he or she prescribes a test $because -$

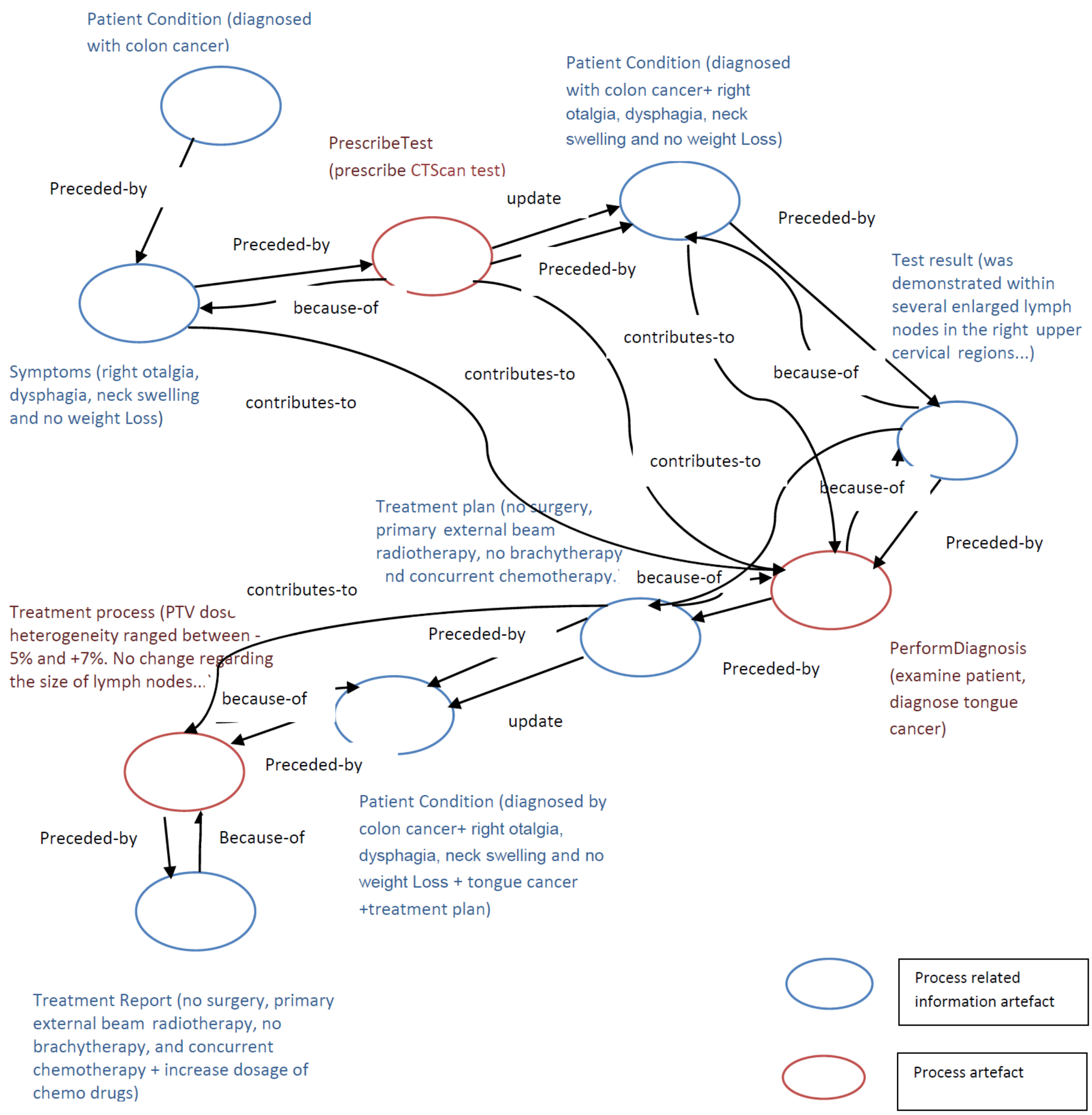

Figure 4.3: Relationship types in a CA-Net

of the observed symptoms. This then results in creation of this type of relationship between $< prescribeTest, Symptoms >$. Another example is when the artefact *performDiagnosis* is generated as a result of an observed *TestResult*, thus relation-

ship type $because - of$ is generated between the two artefacts $< performDiagnosis, TestResult >$.

This relationship type can assist clinicians to find answers to questions such as “Why was this test performed?” or “Why was this diagnosis is made?”. In figure 4.3, an artefact with the type diagnosis is generated because of the artefact with the type test result. A treatment plan artefact type is subsequently generated because of the Diagnosis. This pattern is followed for other artefacts in the CA-Nets. The result of this process are a set of rules in the KB which specifies when an artefact of type Diagnosis exist the Treatment plan artefact relates to this artefact with the relationship type: because-of.

Consequently, this pattern of finding relationship types also generates and completes the rules in the knowledge base as more instances of CA-Nets are observed. For example, if in 100 different CA-Nets there exists this type of relationship between the *performDiagnosis* and *TestResult* then a rule would be added to the knowledge base which relates any *TestResult* artefact type with a *perormDiagnosis* artefact type. This same relationship will then apply in the associated CA-Nets. But for this purpose, each individual *TestResult* must be related to the specific *performDiagnosis* artefact that triggered the creation of that *TestResult*, and not simply to all artefacts with type *performDiagnosis*. This can be achieved when we do consider the semantic annotation of the two artefacts which relates and identifies the conceptual knowledge for a particular artefact.

As an example in figure 4.4, after the doctor prescribed a CT scan of the patient’s tongue, the patient condition is later updated, and the patient is referred to another doctor because of a heart attack. As a result of this second referral, another CT scan test is prescribed for the patient. When the test result for the CT scan of the patient’s heart is generated, then we need to identify the related *performDiagnosis* that trig-

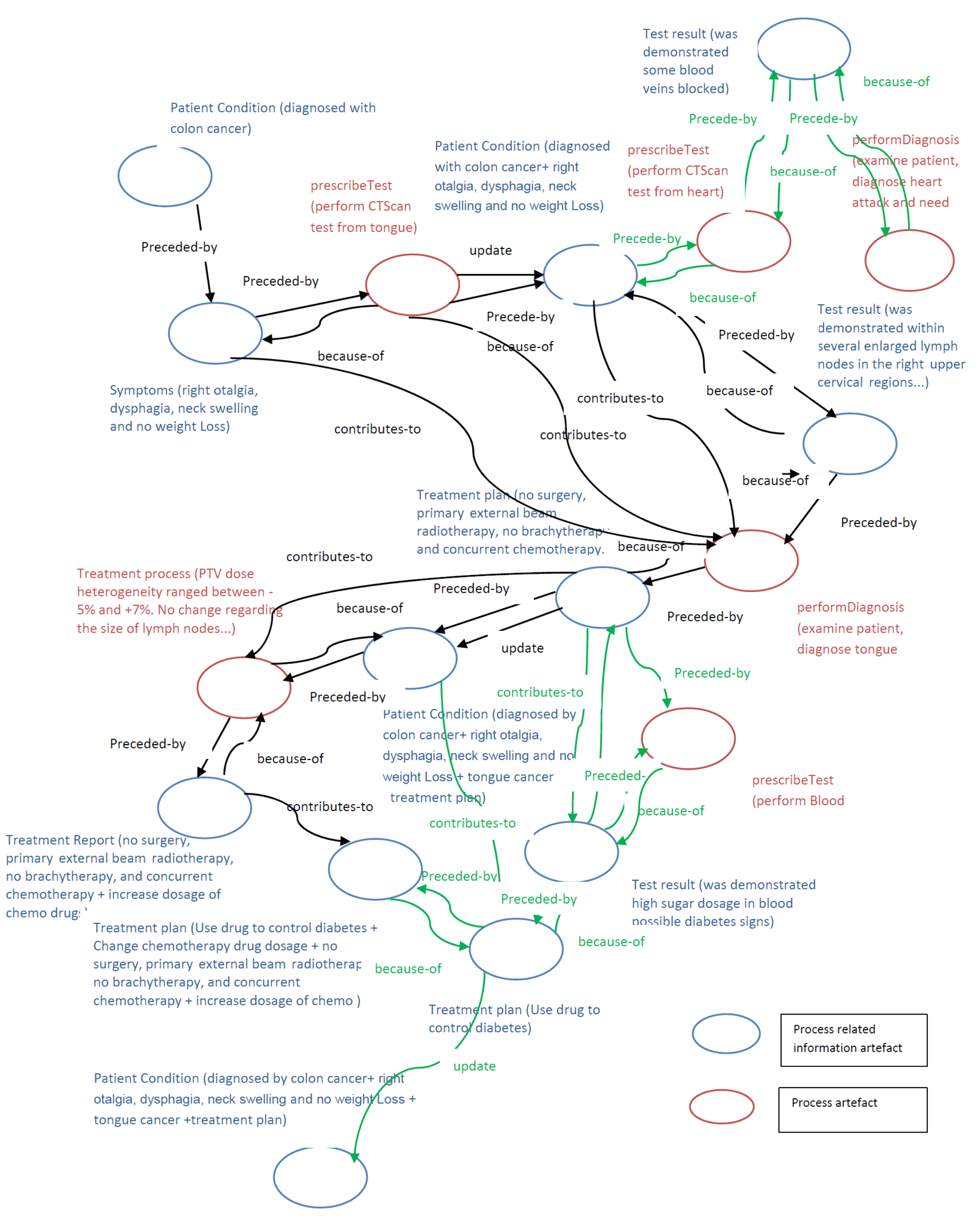

Figure 4.4: CA-Net example for a patient with concurrent treatments

gered this test result. In this case, we need to consider the semantic annotations of the process to be able to find the appropriate artefact that relates to the new artefact. Likewise, when an artefact $performDiagnosis$ is generated, the related $TestResult$ is identified based on the semantic annotation of the existing artefact which is semantically related to the new artefact. In figure 4.3 the two concurrent treatment processes are specified using two different colours to make identifying them easier.

This type of relationship relates any existing and new artefacts, in the case where the new artefact is produced based on the existing artefact. This relationship type is identified between any two artefacts, where there is a need to answer queries such as "Why this artefact is created?". Each artefact type and how they can relate to other artefacts via the relationship type $because-of$ is discussed below.

- Patient condition: This type of artefact can be related to an artefact type such as $PrescribeTest$ which is produced as a result of a patient condition, as a clinician would prescribe a test based on their observation of the patient's condition. Or it can be related to $performDiagnosis$ if a diagnosis can be made following an observation of the patient's condition. The relationship type $because-of$ links the subsequently created artefact (for example, $PrescribeTest$ or $performDiagnosis$ to the existing artefact $Patientcondition$.
- Symptoms: This type of artefact can be related to the types of artefacts which are produced as a result of patient symptoms such as $PrescribeTest$, as prescribing a test is performed based on the patient's symptoms. It can be related to $performDiagnosis$ if a diagnosis can be made following an observation of patient's symptoms. Therefore, the relationship $because-of$ links the subsequently created artefact (for example, $PrescribeTest$ or $performDiagnosis$) to the existing artefact $Symptoms$.
- prescribeTest: This type of artefact can be related to the artefact type which

is produced as a result of a *prescribeTest* such as *performTest* or triggers the creation of a *prescribeTest* artefact such as *Symptoms* or *patientCondition*. It is possible when adding a new artefact *prescribeTest* to a CA-Net, that more than one *Symptoms* or *Patientcondition* artefact exists. Thus, in order to be able to identify the related artefact, we need to consider the semantic annotation of the artefact to find out which of the existing artefacts has the same context as the new artefact.

– Test result: This type of artefact can be related to the artefact type which is produced as a result of a *TestResult* such as *performDiagnosis* or triggers the creation of a *TestResult* artefact such as *performTest*. It is possible when adding a new artefact *TestResult* to a CA-Net, that more than one *prescribeTest* or *performDiagnosis* artefact exists. Thus, in order to be able to identify the related artefact, we need to consider the time stamp and of the artefact and semantic annotations to find out which of the existing artefacts has the same context as the new artefact.

– performDiagnosis: This type of artefact can be related to the artefact type which is produced as a result of a *performDiagnosis* or triggers the creation of a *performDiagnosis* artefact such as *TestResult*. It is possible when adding a new artefact *performDiagnosis* to a CA-Net, more than one *TestResult* or *Symptoms* artefact exists. Thus, in order to be able to identify the related artefact, we need to consider the time stamp of the artefact and semantic annotations to find out which of the existing artefacts has the same context as the new artefact.

– Treatment plan: This type of artefact can be related to the artefact type which is produced as a result of a *Treatmentplan* or triggers the creation of a *Treatmentplan* artefact. It is possible when adding a new artefact *Treatmentplan* to a CA-Net, that more than one *performDiagnosis* or *Symptoms* artefact exists. Thus, in or-

der to be able to identify the related artefact, we need to consider the time stamp of the artefact and semantic effect annotation to find out which of the existing artefacts has the same context as the new artefact.

- Treatment process:This type of artefact can be related to the artefact type which is produced as a result of a
  Treatmentprocess or triggers the creation of a *Treatmentprocess* artefact. It is possible when adding a new artefact *Treatmentplan* to a CA-Net, that more than one *Treatmentplan* or *PatientCondition* artefact exists. Thus, in order to be able to identify the related artefact, we need to consider the time stamp and semantic annotation of the artefact and semantic annotations to find out which of the existing artefacts has the same context as the new artefact.
- Treatment report:This type of artefact can be related to the artefact type which is produced as a result of a *TreatmentReport* or triggers the creation of a
  TreatmentReport artefact. It is possible when adding a new artefact *TreatmentReport* to a CA-Net, that more than one *TreatmentPlan* or *PatientCondition* artefact exists. Thus, in order to be able to identify the related artefact, we need to consider the time stamp of the artefact and semantic annotations to find out which of the existing artefacts has the same context as the new artefact.

- contributes-to: This type of relationship exists when one artefact is related to another artefact to complete the available knowledge or contributes to the generation of the other artefact. Thus, this relationship type is generated when a new artefact is related to an existing artefact but is not generated consecutively directly after the other artefact. For example, in figure 4.3, the artefact patient condition which shows the updated patient health record, contributed to the artefact diagnosis. As a diagnosis usually is made based on several factors such as patient symptoms, patient condition and/or test results, this type of relation-

ship shows the artefacts that in some way contributes to the new artefact. This type of artefact can relate artefacts which are not consecutive, but one artefact contributes-to the generation of the other artefact. To find the existing artefact which is related to the new artefact, semantic annotation, time stamp and the type of queries which need to be answered must be considered.

- Patient condition: *PatientCondition* can be contributes-to artefacts which complete or provide available knowledge to perform a another process, artefacts such as *prescribeTest*, *performDiagnosis*. Before a doctor prescribe a test he or she reviews patient's condition.
- Symptoms: *Symptoms* can contribute-to artefacts such as *prescribeTest*, and *performDiagnosis*, as before a doctor prescribes a test he or she will review the patient's symptoms.
- prescribeTest: *prescribeTest* can contribute-to artefacts such as *performDiagnosis* and *TestResult*. Before a doctor prescribes a test, he or she reviews previous test results.
- Test result: *TestResult* can contribute-to artefacts such as *perdormDiagnosis* and *Treatmentplan*. Before a doctor performs a diagnosis, he or she will review existing test results.
- performDiagnosis: *performDiagnosis* can be contribute-to *PatientCondition* and *Treatmentplan*. Before a doctor performs a diagnosis, he or she will review the patient condition profile.
- Treatment plan: *Treatmentplan* can contribute-to *TreatmentProcess* and *Treatmentreport*. Before a nurse performs the treatment he/ she reviews the treatment plan.
- Treatment process: *Treatmentprocess* can contribute-to *TreatmentReport* and *Treatmentplan*. Before a nurse performs a treatment he or she will review the

treatment reports.

- Treatment report: *TreatmentReport* can contribute-to *Treatmentprocess* and *Treatmentplan*.

- update: This relationship exists when one artefact results in the update of another artefact state. For example, each time the patient visits the doctor as a result the patient health record is updated, and any symptoms or prescriptions are recorded in the patient health record. We can say this artefact exists when a document such as a patient health record is updated. For example, in figure 4.4, after prescribing *performTest* and referring the patient to undergo the CT scan test, the patient condition record is updated. This type of relationship exists between any artefact and patient condition record artefact as it the patient record needs to be updated after every patient referral, test or treatments.

  - Patient condition: The relationship type *update* exists for this type of artefact, every time *patientCodition* is updated.

    The artefacts listed below can have this type of relationship with the artefact *patientcondition* whenever they result in update of a *patientCondition* artefact.

  - Symptoms
  - Test result
  - performDiagnosis
  - prescribeTest
  - Treatment plan
  - Treatment process
  - Treatment report

4.3 Acquiring Information using CA-Nets

As mentioned earlier, CA-Nets provide a layer between data sources and required queries. CA-Nets make the process of obtaining and extracting patient specific treatment easy. They support various types of queries which are defined below.

4.3.1 Temporal Context

The first type of query is to find a particular most recent artefact in the CA-Nets. These queries are useful when a specialist wants to justify a decision or review a patient's history.

For example, in Figure 4.5, when a specialist needs to justify to a GP their previous diagnosis for the patient, he or she needs to track back through the artefacts to find the diagnosis artefact. A CA-Net can be used to answer queries such as "What is the most recent diagnosis?" or "What is the nearest previous patient condition record?". To answer such queries one must back track through the CA-Nets for a specific patient and find the first "diagnosis" or "patient condition" artefact. For example, when a patient who is diagnosed with a tongue cancer is referred to a specialist, the specialist will review the patient's history as he or she will want to know the details of the previous test result which resulted in the current diagnosis. In order to find this information, the doctor can start from the diagnosis artefact and follow the relationships to each of the related artefacts to eventually find the reasons behind the current diagnosis.

4.3.2 Causal Context

Another type of query that could be answered using CA-Nets is a query that relates to finding the reason behind a particular artefact. For example, we can answer the following queries: "Why was this blood test prescribed?" or "Why did the doctor prescribe this drug?". In these cases, one must backtrack through the CA-Nets from the specific artefact to find the

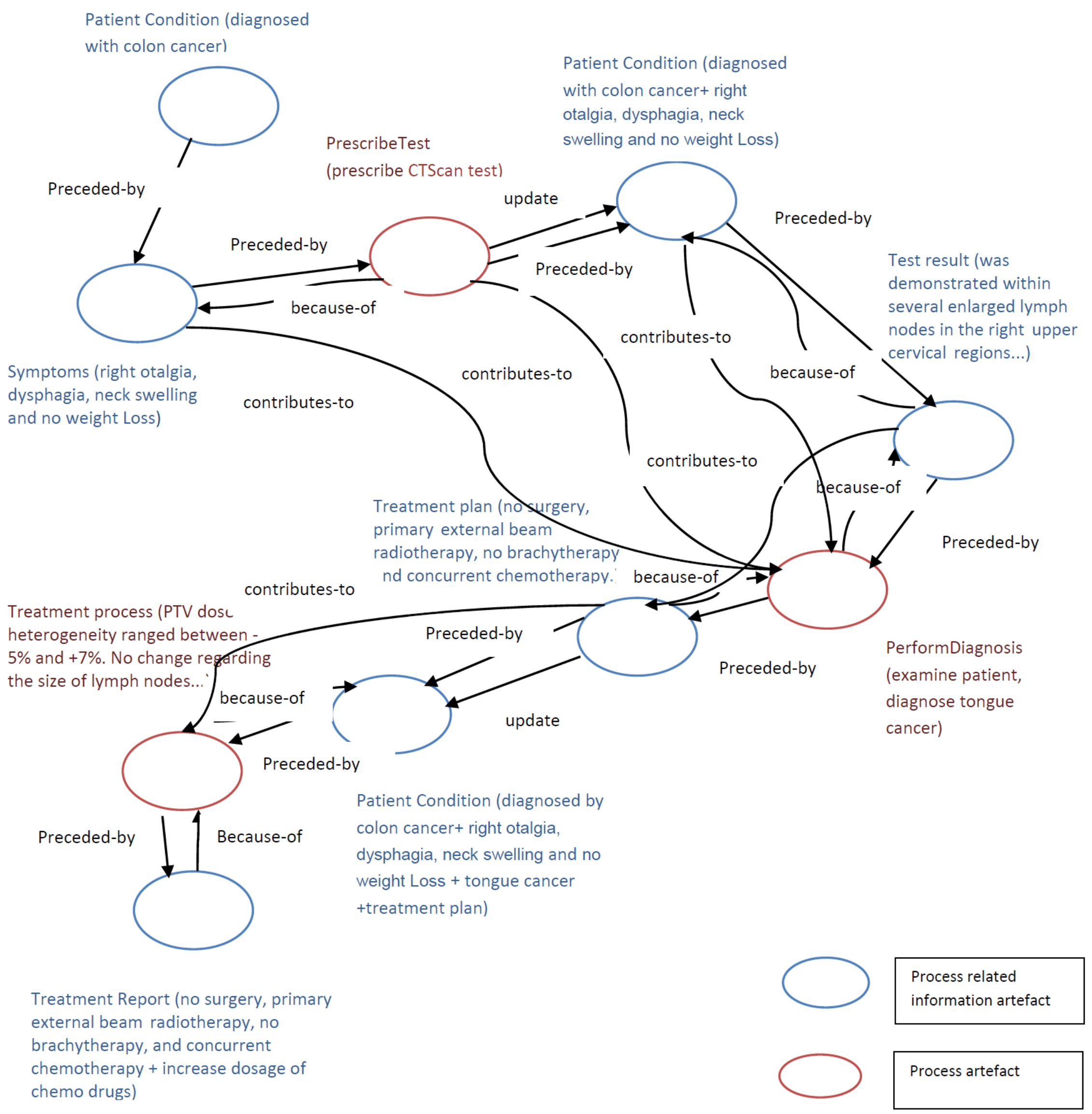

Figure 4.5: CA-Nets for a patient

associated reason. Moreover, it enables clinicians to ask questions in a sequence format and follow the reasons behind every related artefact. To illustrate this further, suppose we first ask " Why was this patient diagnosed with tongue cancer?". As we follow the diagnosis artefact in figure 4.5, we find that the diagnosis was done based on the CT scan report". Then we can ask "Why was the CT scan requested?" and then backtrack to the artefact test

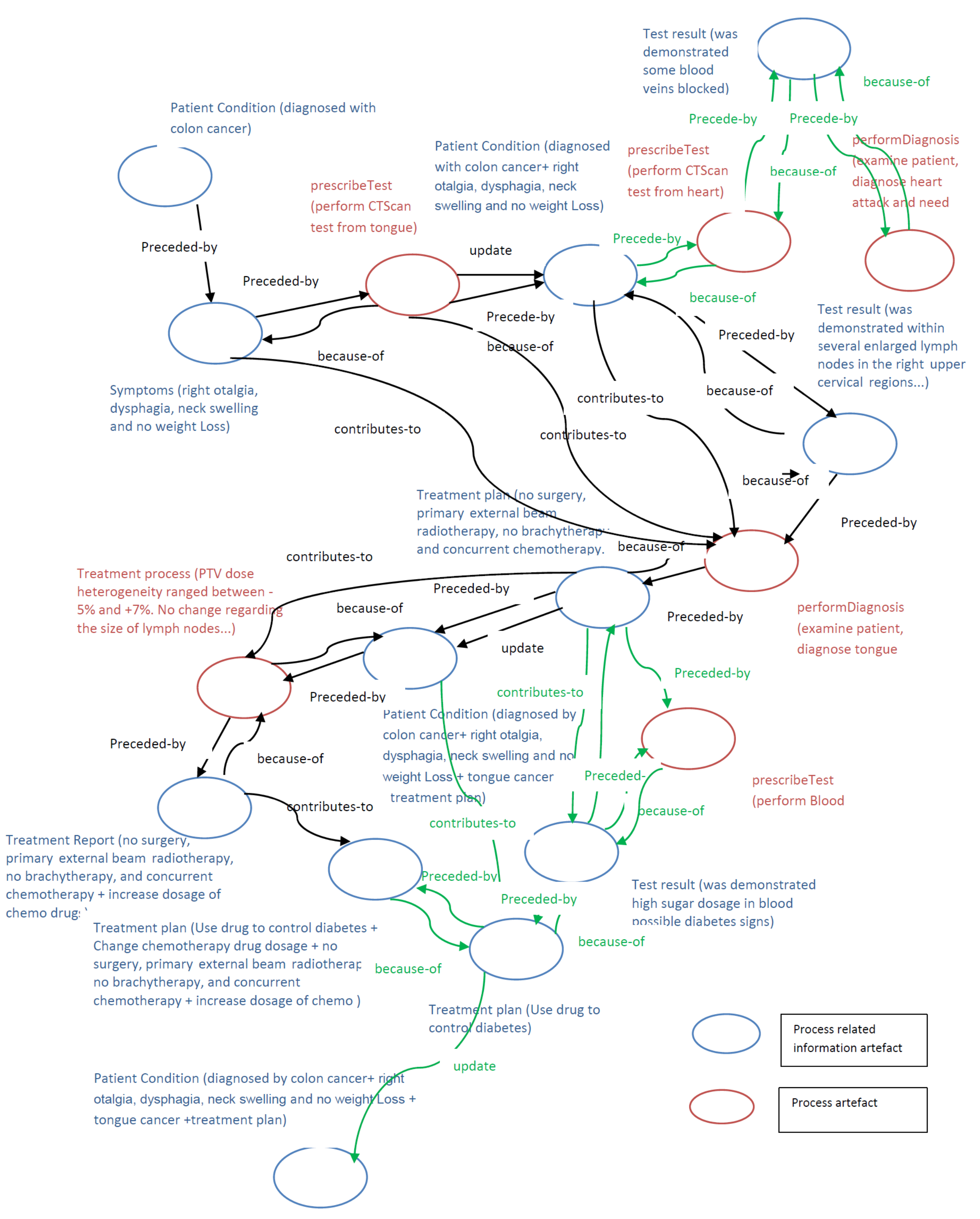

Figure 4.6: CA-Nets for a patient with concurrent illnesses

and find the answer because-of the patient's symptoms.

This feature is particularly beneficial for clinicians when they need to justify previous decisions.

4.3.3 Responsibility Context

Information regarding the owners or the performers of the processes is another critical query. The process owners in the context of healthcare systems, include doctors, nurses and clinicians. To obtain such information, we would simply need to refer to the annotations associated with the specific artefact. The information in artefact annotations specifies the owner for each artefact.

4.4 Reasoning over process effects using CA-Nets

In section 2.6, the theory of abductive reasoning was discussed. The potential applications of abductive reasoning in CA-Nets are explained using different examples in the following section.

The use of abductive reasoning is explained using example 3.3.1. In healthcare systems when a patient is referred to a doctor, based on the observations (O), which are the patient's symptoms and condition, the doctor hypothesizes that certain diseases may have caused those symptoms. To prove the proposed hypothesis, the doctor prescribes a test(s) and chooses the explanation (E) which is confirmed by a combination of the test results and the existing facts (T). In example 3.3.1, the patient's symptoms are right otalgia, disphagia, neck swelling and no weight loss. The doctor proposed hypothesis is tongue cancer. To prove this hypothesis the doctor prescribes some examinations or tests and then chooses the hypothesis (or narrows the options down to two or more hypotheses) that is confirmed by the test results [1].

We now explain how the rules in knowledge base can be completed with the use of abductive reasoning.

KB is a domain specific knowledge base that describes the clinical facts as a set of rules written as a knowledge dictionary(as explained in section 4.1). The rules in the knowledge base evolve as more observations are gleaned from different CA-Nets. As discussed in chapter 3, the KB is incomplete and the rules are completed as new observations are made and new rules are derived from the new observations and subsequently added to the KB. For example, we observe that when process α occurs document β is generated, or whenever document β is generated then process γ should be observed. Then these rules are applied to the KB.

As the rules in the KB are completed, they can be used in the process of generating CA-Nets. In a CA-Net, if an artefact of type α is found, one would expect to have an artefact of type β afterwards. So, if artefact β, was not found, then a flag can be raised to indicate a discrepancy. In this way, the execution of the process model can be verified, or discrepancies found between what is actually happening and what was expected to happen.

Effect annotations are critical for applying the reasoning to the processes as they provide the required annotations for each task in a process. Thus, we need to provide a comprehensive knowledge base which supports this purpose by matching cumulative effect annotations with the rules in the knowledge base. For a given ordered pair of tasks with effect annotations, determines the reason after both tasks have been executed. We assume throughout, the existence of a background knowledge base that provides an additional basis for reasoning. By matching the cumulative effect annotations of each process and the rules in the knowledge base in the same format, we can support reasoning applied to processes.

The use of abductive logic in CA-Nets, is explained using a real scenario that was investigated in one of the local cancer care centres. A patient with a cancer diagnosis was under treatment with a chemotherapy protocol. Before each session of chemotherapy, the

patient needed to undergo a blood test and be examined to determine whether or not to continue the same chemotherapy plan. Based on the symptoms and test results, clinicians may change the plan, for example, by changing the drug, the drug dosage, the treatment period, etc. In each session, a particular patient may be treated by any of the available nurses who need to document any required deviation in performing the protocol plan, such as any change in the drug dosage, drug type, cycle change, etc. These documents are useful for the clinicians who need to evaluate the treatment plans and, if required, change the plans for a better result. What actually happens in a real work environment is that, documentation is often missing as a result of time or resource shortage or negligence. This makes the job of decision justification difficult.

CA-Nets based on abduction reasoning can be used to solve this problem. In the scenario where a clinician needs to manage a protocol plan for a particular patient, she or he will check the previous sessions of the plan, and find that the drug dosage and treatment date were changed (O), without any documentation explaining the reasons behind the change. Thus, the clinicians will propose some possible hypothesis (H), for example, the drug dosage modification for a chemotherapy protocol could be as a result of a high level of toxicity shown in patient's blood test, or the patient's reaction to the drug during the plan, or the drug's interaction with other types of drugs prescribed to the patient, or detrimental effects on different body organs, etc. To find the best possible explanation (E), the clinician will prescribe a blood test for the patient. Based on the blood test results, the clinicians will select the best explanation as to the reason for the drug dosage change. As a result of this computation, a new artefact of type Reason which indicates the reason for the observed deviation is generated in the CA-Nets.

Furthermore, this will result in a new rule being applied into knowledge base. The new rule indicates that when artefact α is observed with shows an annotation indicating discrepancy, an artefact reason should be subsequently generated.

When producing CA-Nets for this particular example, each time the patient is referred to the Cancer Care Centre for treatment, several artefacts are generated. Based on the rules in the KB, which are defined in both informal and formal logic, and the proposed algorithm, the CA-Nets is created. In the KB, a rule specifies that before starting a treatment protocol, a patient needs to undergo a blood test: Perform (Patient1, bloodTest). Thus, we first expect to have an artefact of type a test, which is a blood test report, and as defined previously each new artefact will need to have some attributes such as artefact type, owner and effect annotation. If the artefact is a document it may not need any annotation, so an artefact of type test and a blood test attribute will be created. On finishing the protocol procedure, another artefact will be generated with type treatment plan. The person who has performed the protocol, for example, Nurse1, is the owner of the artefact and the cumulative effect annotation for the process is created. Based on the KB, the appropriate set of relationships will be selected and created. And thus the procedure continues. For the purpose of deviation analysis, each pair of cumulative effect annotations is compared to find out if any deviation has occurred. If a deviation has occurred then the KB will be searched for any rules that could support the observed deviation. For the purpose of this case study, we have generated a set of rules based on the huge data log to which we have access. This data log consists of the data from about 12 different protocols for treatment of about 200 patients. Each protocol is performed a different number of cycles for different individual patients. The number of cycles specifies the number of times that the protocol is performed for the patient. It varies as the patient condition may change during the treatment protocol, because of disease progression or unacceptable level of toxicity or even patient death. By analysing this huge record of data, we can create the business rules. The rules can be supported by the rules in the eviQ information system and also confirmation by one of the radiation oncologists from a local cancer care centre. Then, based on the generated rules, and abductive reasoning we can provide the best possible reason for any deviation in the system.

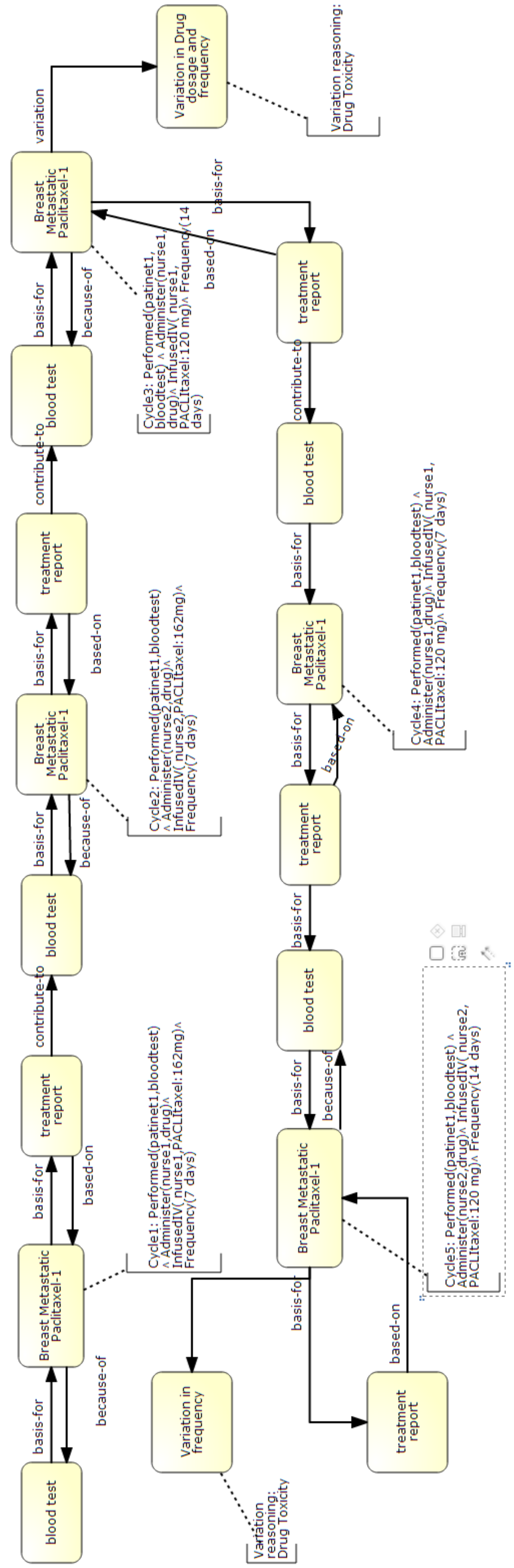

Figure 4.7: Reasoning over process effects

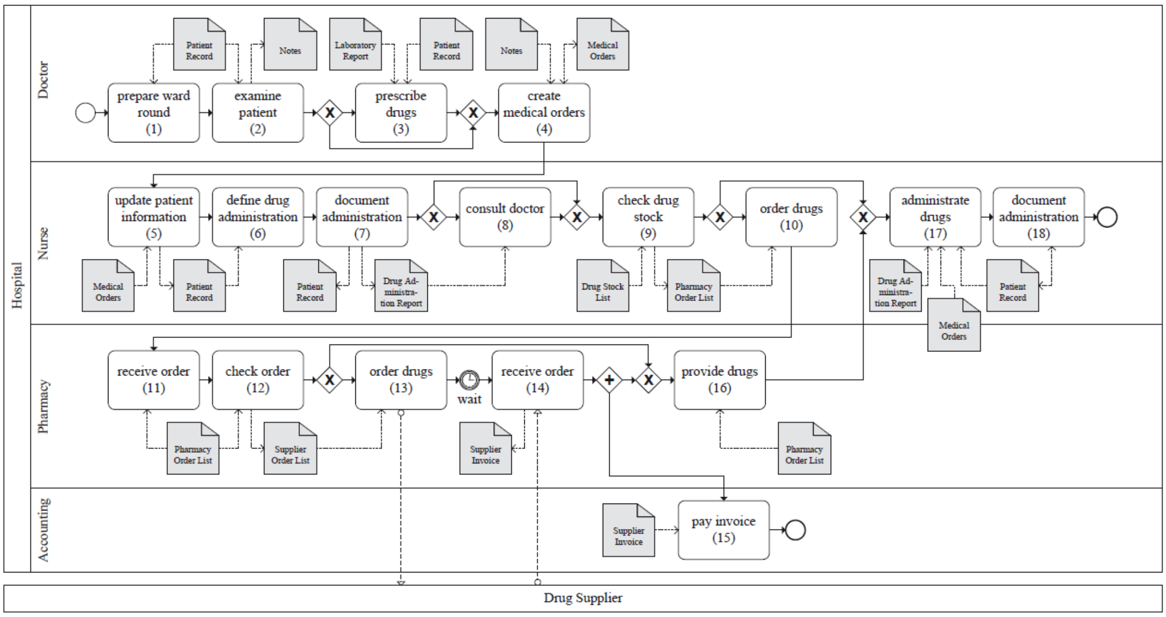

Figure 4.8: Procurement of drug, [44]

In figure 4.7 when a variation is observed, based on the rules and abductive reasoning the CA-Nets generate another artefact which indicates the best possible reason to explain the deviation.

4.5 Comparing CA-Nets with POIL

This section provides a comparison between CA-Nets and Process-oriented Information Logistics (POIL) which was discussed in section 2.7. The aim of POIL is to provide the process-oriented and context-aware delivery of process-related information for knowledge-workers. Michelberger, Mutschler, and Reichert [44]proposed POIL to bridge the gap between business processes and process-related information.

As it was demonstrated in chapter 2, in a semantic information network (SIN), all the business process objects in a process model such as tasks, data objects, gateways, etc. and corresponding process instances transform into a node in a SIN. The resulting SIN from the process model in figure 4.8, is demonstrated in figure 4.9. This figure shows that each task,

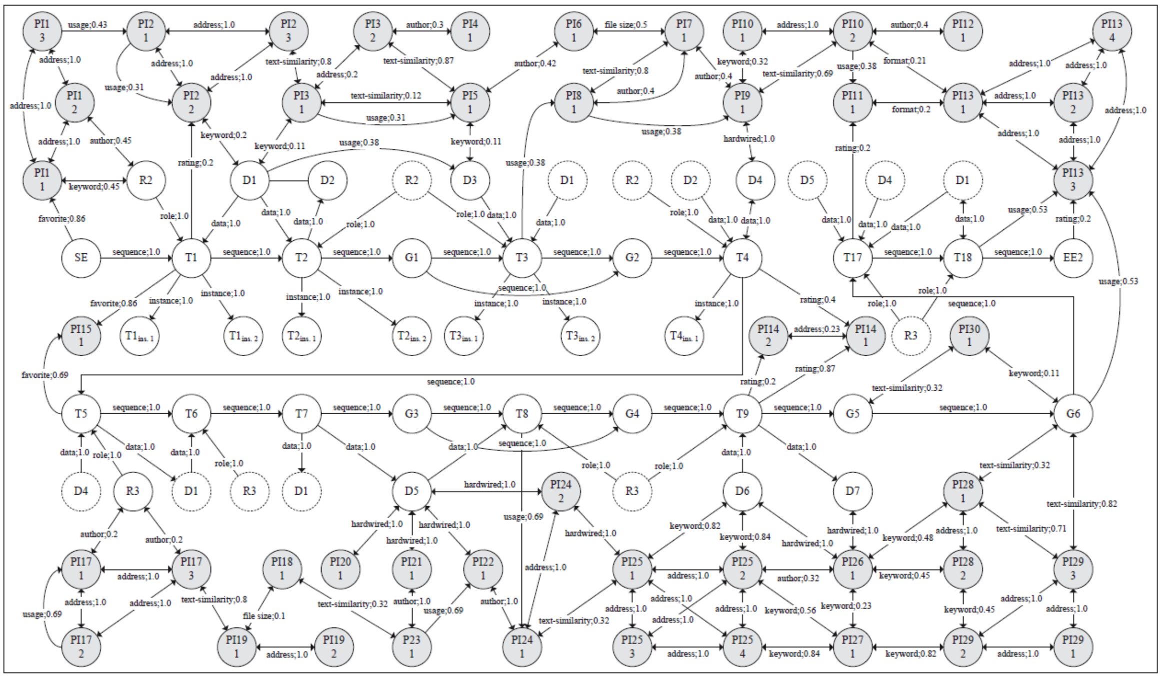

Figure 4.9: Final SIN, [44]

data item, object, owner, and gateway compromise the nodes in a SIN and the relationship objects transform into edges between the process nodes. Whereas in CA-Nets, we assume each process instance (at an abstract level) or task (at a more detailed lower level) to be a new artefact (node) in the network. The type of each artefact specifies the appropriate edge type between any two artefacts.

In POIL, process objects like role or pool, are also assumed to be nodes in a SIN, but in CA-Nets they are specified as node annotations. Using effect annotation is another capability of CA-Nets which enables reasoning over processes and other motivations such as process monitoring. This method results in the elimination of unnecessary nodes in the network which helps in knowledge representation. As discussed earlier, one of the main goals of CA-Nets is representing the knowledge to assist knowledge workers to find the required knowledge faster and easier.

Reasoning, which is discussed earlier in detail, is another purpose of CA-Nets. This ca-

pability is another difference between these two approaches. In POIL, each edge is labelled by relation reason and relation weights, but it doesn't explain how they provide reasoning process. In figure 4.10, we demonstrate how we obtain CA-Nets from the process model based on the example in figure 4.8 [44]. It demonstrates that every process and process related information from the process model in figure 4.6 is transformed into an artefact in a CA-Net. In figure 4.10, the execution of the first three processes of the process model in figure 4.8 are showed below of that. It illustrates after execution of prepare ward round process, according artefact is generated in a CA-Nets. Following to the process patient record is reviewed, which results creation of an artefact patient record in a CA-Net. Then examinePatient process is executed which results in generating the according artefact in the CA-Nets. Then prescribeDrug process is executed and creates the according node in the CA-Nets. As any process executed, it results in generation of the according artefact in CA-net with the effect annotation and with generation of the next artefact the effect annotation is cumulated for the new artefact. In figure 4.11 the execution of the process model is illustrates with the artefacts(nodes), relationships (vertices) and corresponding cumulative effect annotation is demonstrated.

In POIL, in addition to the SIN, a context model is constructed based on available context objects (this is discussed in detail in [44]). A context model is an ontology-based model and uses pre-defined context factors such as user, location, device or time. The context model enables the representation of all context information that is relevant in the current situation of a process participant, which can then be used to filter the SIN [44]. The context model is completely independent from the SIN, and context objects are only stored in the context model. Hence, there exists a specific context model for each user, but a central SIN for all users [29]. The existence of the context model made the SIN a very useful framework. It can support queries with different type of vertices.

In our research, CA-Nets are used to represent knowledge about a patient specific treat-

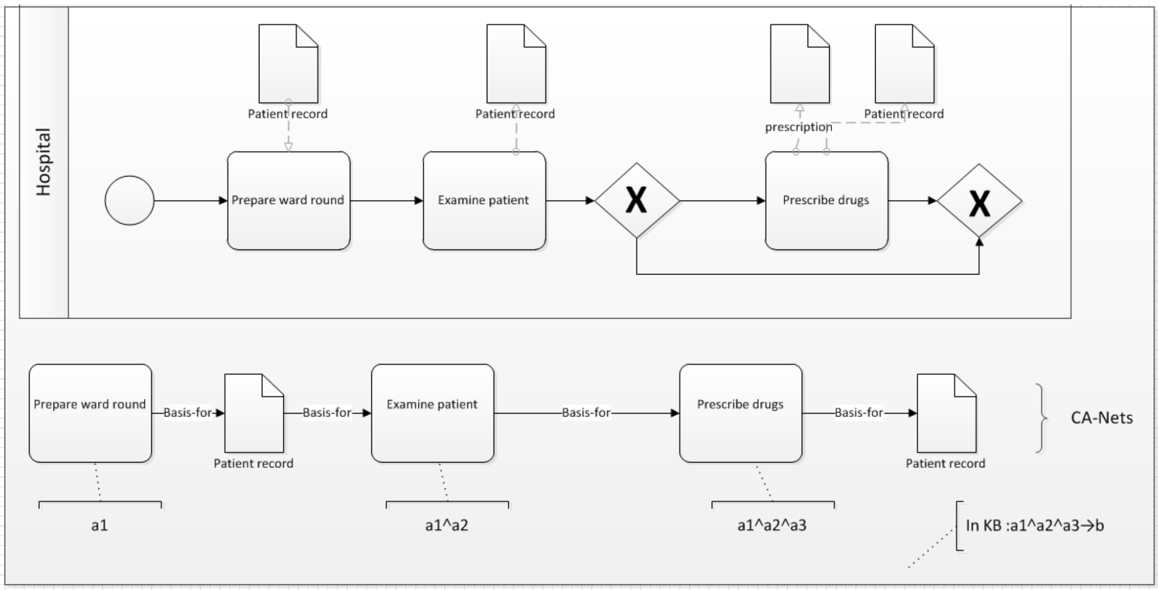

Figure 4.10: CA-Nets

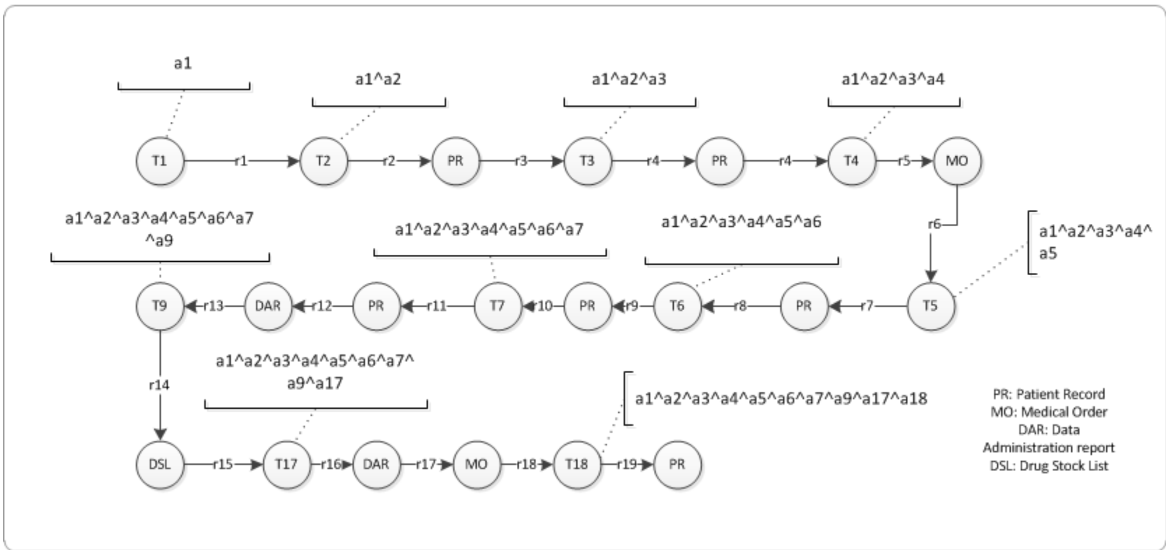

Figure 4.11: Generated CA-Nets

ment and can support a limited range of queries.

4.6 Conclusion

This chapter discussed the need for data acquisition in a related and meaningful format, in order for specific data to be assessed easily and used by different users with different information needs in healthcare system settings. Today's disparate data overload makes it difficult to provide clinicians and decision makers with the information they need in the right format and granularity. Focusing on this need, a context framework to help solve this problem was presented and the procedure of crating a CA-Net is discussed using different examples.

The theory of abductive reasoning was discussed and potential applications of abductive reasoning in CA-Nets were explained using different examples. The use of effect annotations was discussed as a requirement for reasoning and the procedure of reasoning was explained with an example. The last section provided a detailed comparison between CA-Nets and another related approach called POIL.

Chapter 5

CA-Nets in Practice

5.1 Introduction

In Chapters 3 and 4, we have discussed our approach and explained how to represent knowledge with CA-Nets from the available information. Moreover, we discussed how we could apply abductive reasoning to support causal queries. We also described how abductive reasoning can help with completing the KB. In this chapter the architecture of CA-Nets is described. Then we demonstrated the use of CA-Nets with two different case scenarios. The scenarios are based on real case studies in health care systems. The first one is based on the motivation example and the second one is selected from discussions and information gathering from one of the local Cancer Care Centre.

5.2 Implementation

In this section, the architecture of the tool as well as steps taken to generate CA-Nets are explained. The architecture of CA-Nets is displayed in figure 5.1.

Activiti is a business process management platform. It was used to model the workflows. Activiti provides an environment to draw business processes, called Activiti Designer.The

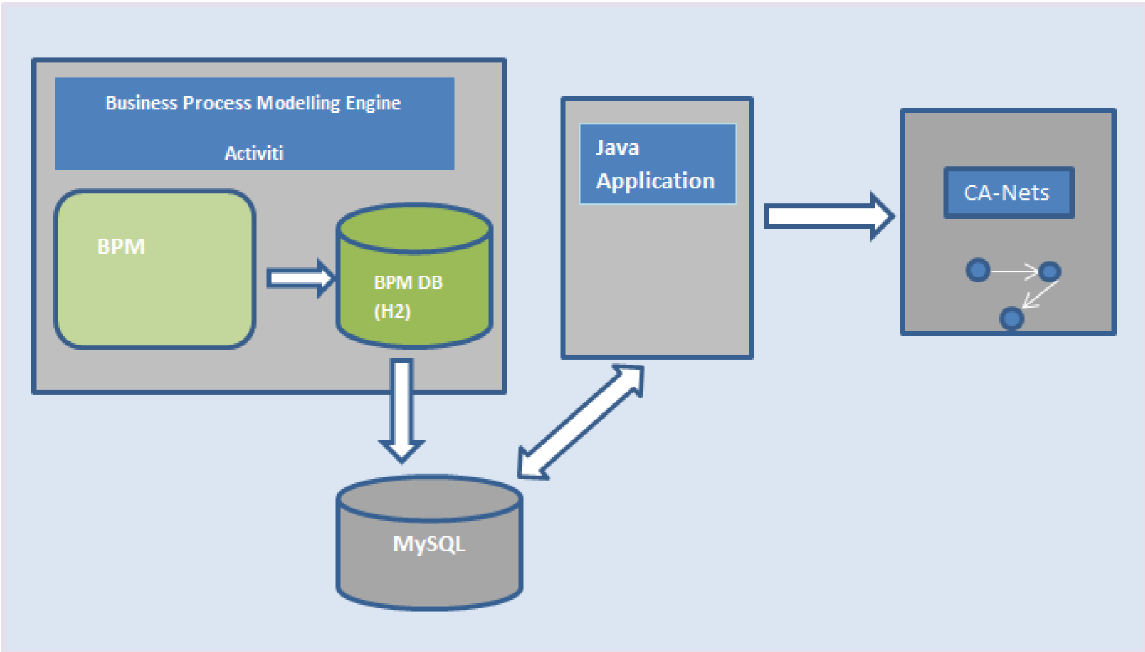

Figure 5.1: Architecture of CA-Nets

XML output of the Activiti Designer is deployed to the Activiti Engine that runs the process definition. A process definition consists of elements like events, tasks, and gateways that are connected together via sequence flows. When a process definition is deployed on the process engine, a process instance is created. The process engine enables, simulating the wokflow. It enables running the process model and as a result the process instances are generated and recorded in the according tables in a databse. The process instances are stored in the Activiti database (H2). In order to browse and use Activiti, Apache Tomcat should be installed. Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies. Tomcat provides a HTTP web server environment in which to run Java code. The Activiti (H2) is a database connection module. In order to access and read the data from H2, we need to configure H2 to connect it to a relational database. We selected MySQL for this purpose. Data was read from the database using Java Eclips. Java source code (provided in Appendix A) was developed to demonstrate the generation of

partial CA-Nets based on two different scenarios.

To represent CA-Nets in practice, we first map the clinical process in Activiti platform. Then we ran the process model for several patient treatment. The Activiti database(H2) was configured to connect to Mysql, in order to be able to select the knowledge for a specific patient via queries. Finally we represent the extracted knowledge in a related graphical format. This way we provide a schema of a CA-Net in practice. We demonstrates how it helps follow patient specific treatment in a healthcare system.

5.2.1 Scenario 1

One of the scenarios that we used to illustrate how CA-Nets are useful, is the process of a GP visiting a patient. We first modelled the appropriate careflow for the scenario which is described as follows. This scenario is based on the motivation example from section 3.2. It starts when a patient visits a GP. The very first thing that the GP needs to know is whether the patient is visiting the GP for the first time or is it a follow up visit. If it is the first time visit then the GP will input the patient's information into the system and ask the patient to describe their symptoms. Then the patient will be examined and be referred to undergo a test or be diagnosed based on the GP's decision. But if the patient is not visiting the GP for the first time, then the GP will review the patient's health record and patient's test results if available. The document is then saved in the system by the GP and the process ends when the GP writes a prescription which could include required drugs or test(s). The corresponding process model is displayed in figure 5.2. It provides the sequence of the tasks as well as the generated related document which is the patient HealthRecord (patientHR).

After deploying the model, we created the required instances to display them later as CA-Nets. We extract the data from the Activiti (H2) database into the mySql database. Then for a particular patient we selected the related processes from the process log and based on the time stamp of each process, we represented the CA-Net in figure 5.3.

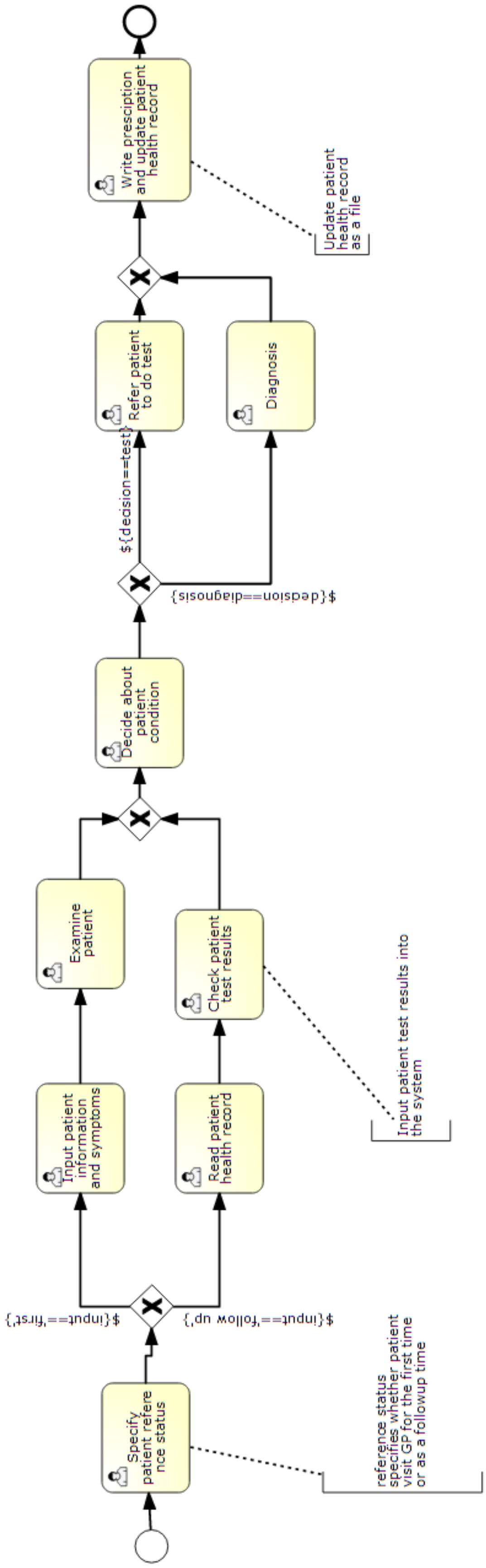

Figure 5.2: Patient visiting a GP

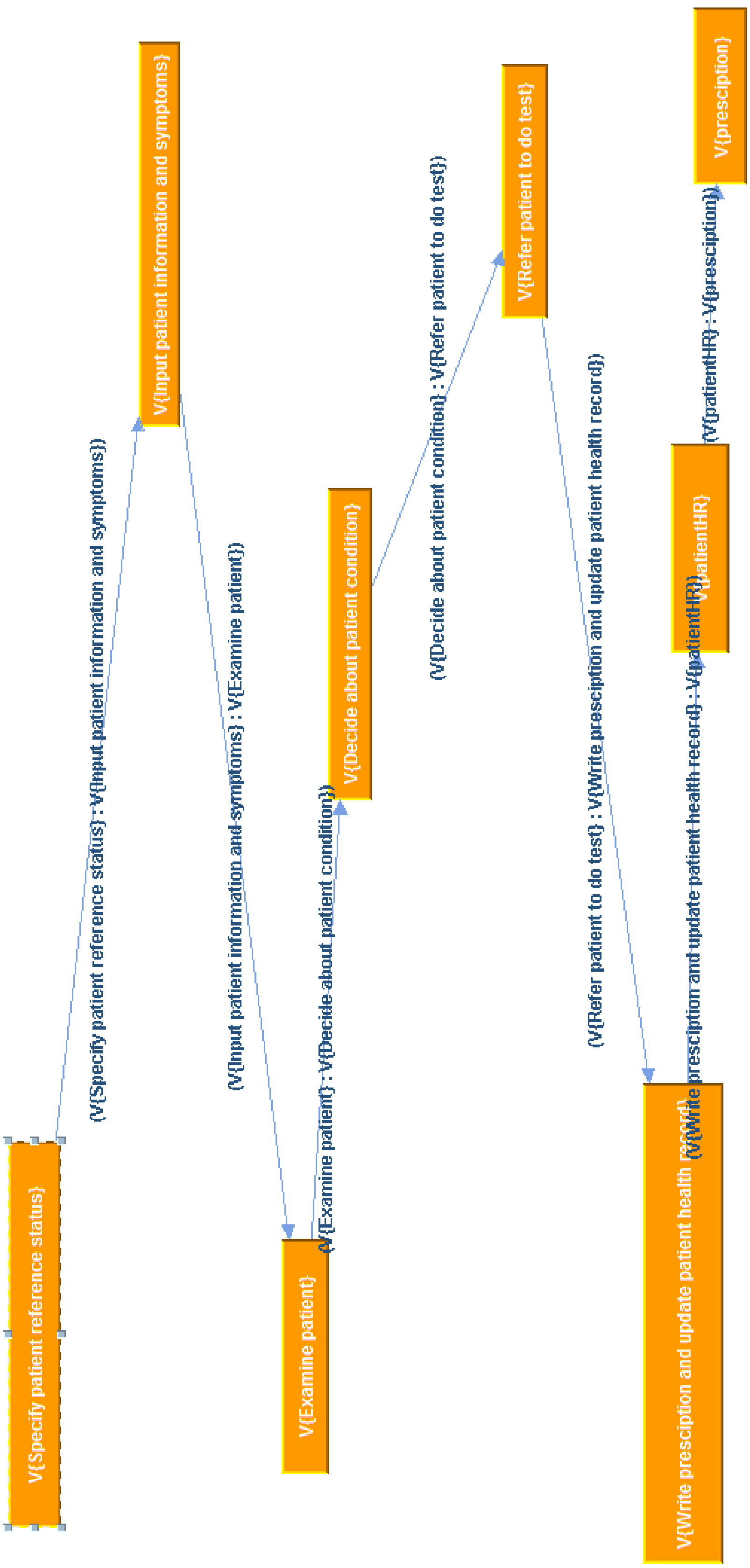

Figure 5.3: Patient visiting a GP.CA-Nets

As shown in figure 5.3, we have the patient history in a graphical format which shows the sequence of activities and presents the related documents such as patient health record, prescriptions, test results, etc. for a particular patient. It assists physicians and specialists to review and find the related activities and documents for a patient in order to evaluate previous actions and decide on the next appropriate activity.

The output of this process model demonstrates the sequence of the executed processes and generated document in a graphical format. This representation assists in finding the answers to the queries that we discussed before in chapter 4. This helps clinicians to find the answer to the temporal queries, by tracing back or forward the artefacts to find about the sequence of the tasks and related documents. Thus, instead of spending a lot of time to read and accessing different systems to find the related knowledge, they could start tracing from any artefact in CA-Nets that they are interested to know about.

5.2.2 Scenario 2

The second scenario is based on a real life scenario. It is based on interviews with one of the radiation oncologists from the local cancer care centre. According to the interviews with the oncologist, we found that oncology specialists use eviQ cancer treatment plans as a standard to define treatment plans for different patients. eviQ Cancer Treatments Online is a clinical information resource that provides health professionals with current evidence based, peer reviewed, best practice cancer treatment protocols and information. All of the content provided complies with an accurate data governance model, and clinicians view the resource as a reliable information system within the Australian context.

In the cancer care centre they use an existing loosely defined template of chemotherapy protocols and modify it to satisfy the patient condition. Deviations from the template might include prescribing a different drug type, changes to the drug dosage, treatment cycle, or adding or deleting some prescribed drugs. As a result, the modified protocols will be saved as a new protocol in the system. Unfortunately, this practice will cause a redundant system full of protocols which may be used only once. Thus, it will be hard to manage the data and find the appropriate protocol.

One of the centre's requirements was to find the common reasons behind the modifications made to the chemotherapy protocols and also the most deviated instances of protocols. They need to know why the protocols deviate when they are used in a real situation, and what are the actual reasons behind the deviations. Furthermore, they need to know the frequency of deviated instances from the standard protocols. This information can be used to update the existing protocols that have the highest rate of occurrence of deviations, in order to reduce the need for protocol modification. With CA-Nets we can assist this problem by displaying the common reasons which cause deviations in a treatment protocol cycle. We generated CA-Nets based on the real data log, which contains data about each treatment protocol for a particular patient, such as protocol name, treatment date, drug dosage, etc.

The generated CA-Nets can help clinicians to be able monitor patient treatment and justify treatment protocol if required. Moreover, they assist protocol designers to monitor any particular protocol treatment for a number of different patients in a real situation. Thus, they are able to adjust protocol designs more precisely based on real situations, which will result in less protocol redundancy.

To achieve this, we first modelled the proposed protocols using the BPMN modelling language. After analysing several protocol treatments, it was found that it is possible to develop a process model at an abstract level which could satisfy all the different types of protocols in eviQ, as shown in Figure 5.4. Although, this BPMN model can be used to show the abstract level of all treatment protocols, we decided to design a more detailed model for one of the protocols for the purpose of clarity. The detailed treatment protocol model, in order to make the process evolution more clear. The cancer care centre provided us a real data log for the different protocols to use in our research. The data log assisted us in the process of deviation reason discovery.

After reading and analysing eviQ treatment protocols, analysing the data log and interviewing the radiation oncologist, we found the main reasons behind protocol deviations. One of the main reasons found was toxicity, which results in deviations in drug dosage. Lack of available resources and treatment cycle dates coinciding with weekends, are the common reasons which cause deviations in the treatment cycle date. The process model was generated taking into consideration the different factors which would cause a deviation whether in drug dosage or in protocol treatment cycle dates. The resultant model is shown in figure 5.5. Then we used the data log from the ICCC, to illustrate the real data using CA-Nets to help identify the deviations that occurred and the reasons behind those deviations. For this protocol, the data for about ten patients during each cycle of their chemotherapy treatment was analysed.

The output for this example is displayed in figure 5.6 to 5.10. It is divided into five

different figures in order to display the content as legible as possible. The data used is from the local cancer care centre data log. In this example the patient was referred to the cancer care centre to undergo a chemotherapy treatment protocol prescribed for breast cancer. The process starts when the patient is visited by the nurse who is responsible to perform the chemotherapy. The nurse specifies the patient reference status, to determine if that is the first treatment or a follow up treatment. Then the patient is undergone a general assessment and if there was no problem with patient condition, then the nurse commences the protocol. The nurse admins the drugs as prescribed with the specified dosage. Thus, she or he admins LORATADINE 10 mg, RANITIDINE 150 mg, and DEXAMETHASONE 3 mg and PACLItaxel for 80 mg. After the protocol conducted successfully, the nurse makes arrangements for the next treatment cycle, does any post treatment required and send patient home. At the end the patient's record is updated. In the first cycle, the protocol was conducted normally, but in the second cycle of treatment a minor level of toxicity was detected which prompted a deviation in the drug dosage. Figure 5.8 shows the beginning of the second cycle, when the patient is referred to a nurse and he or she starts the protocol. In the second cycle, the patient referral status is specified and the patient is undergone a general assessment, but this time the level of toxicity in patient's blood make the nurse to deviate the drug dosage. Thus, the nurse admins the same drugs, but this time reduces the DEXAMETHASONE to 1.5 mg and PACLItaxel to 40 mg. After the protocol finishes successfully, the nurse makes arrangements for the next cycle and does the post treatments if required. At the end the patient's record is updated. By reading and analysing the output we can follow patient treatment cycles and find the deviations and the reasons for those deviations based on real data.

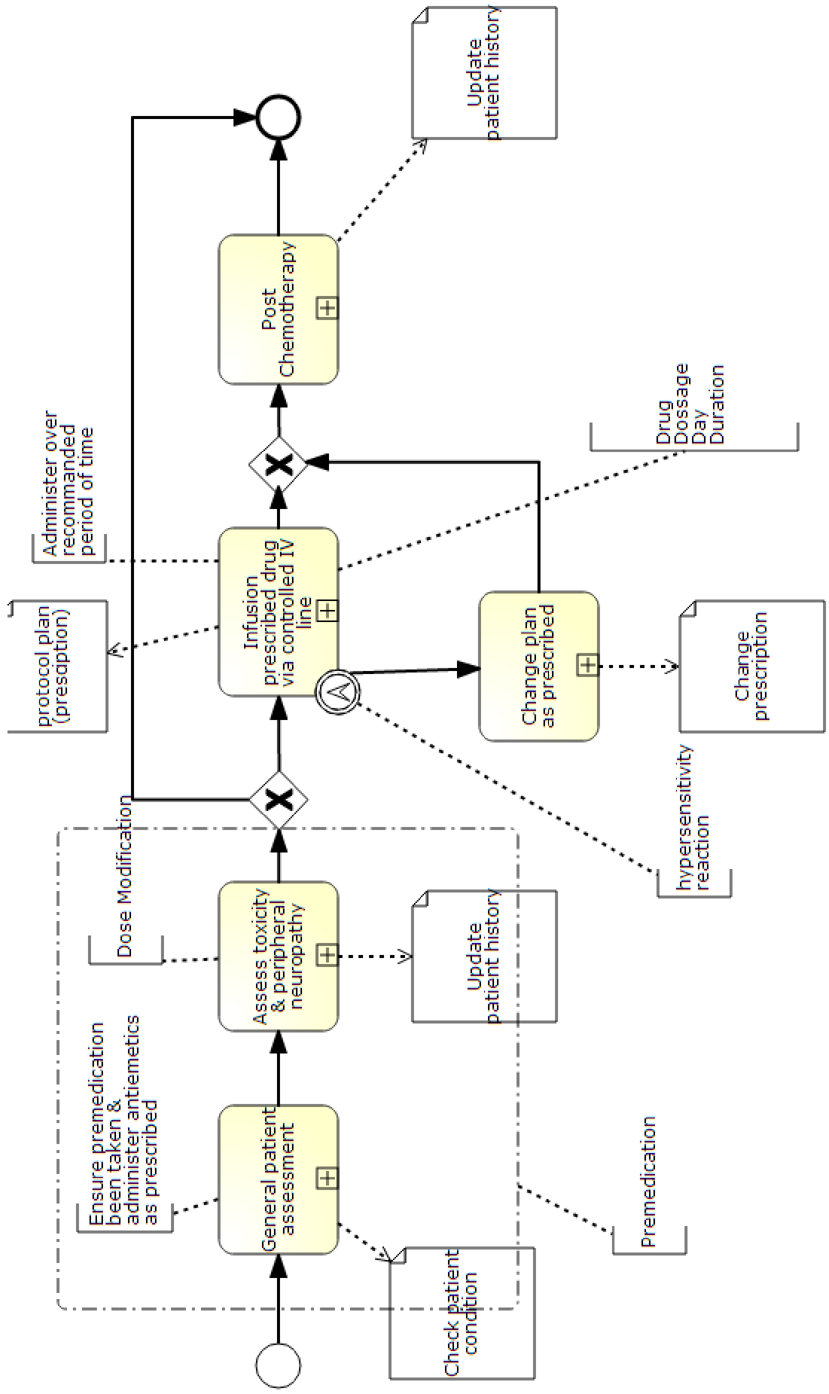

Figure 5.4: Abstract level of treatment protocols

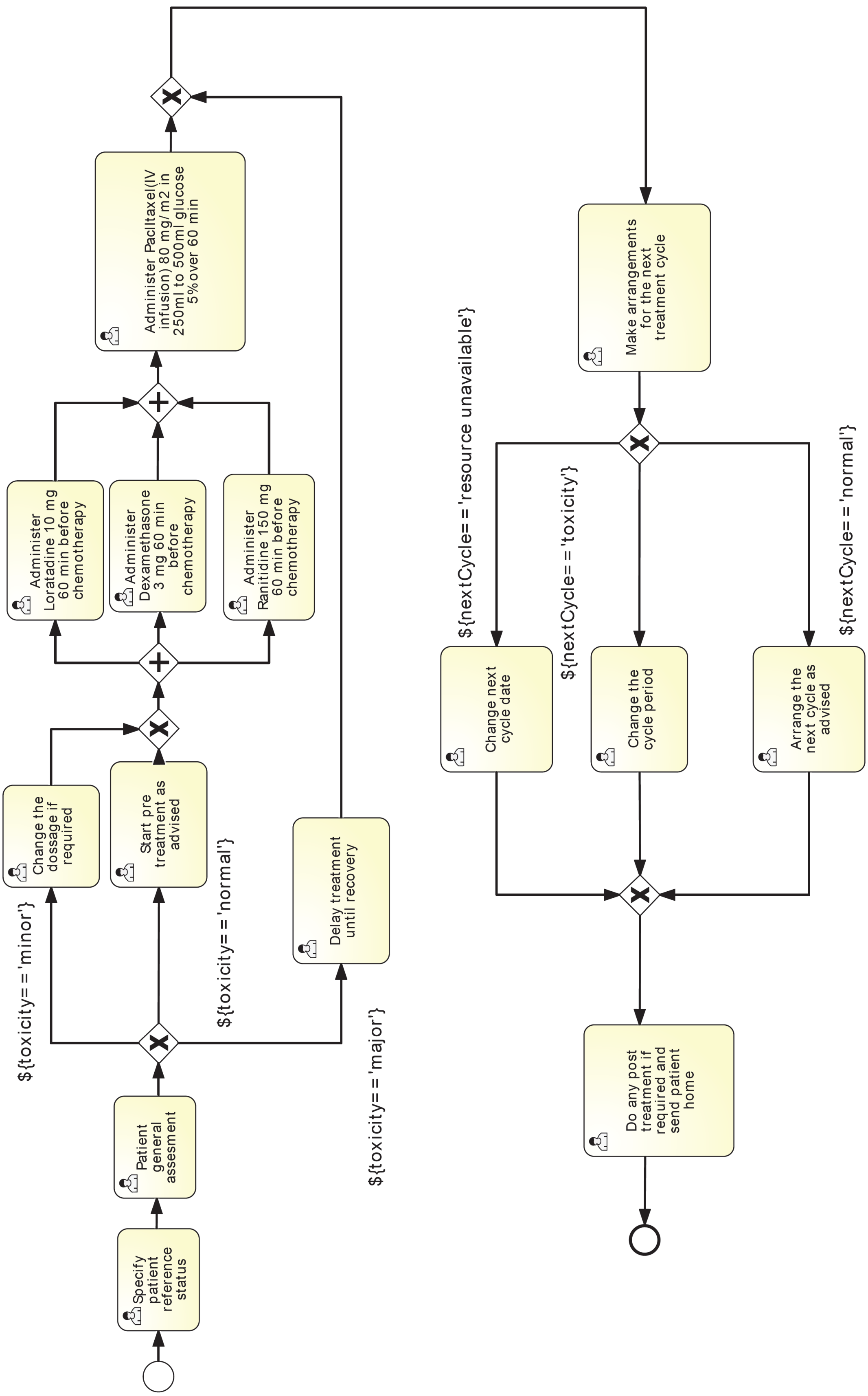

Figure 5.5: Breast Cancer Protocol Treatment

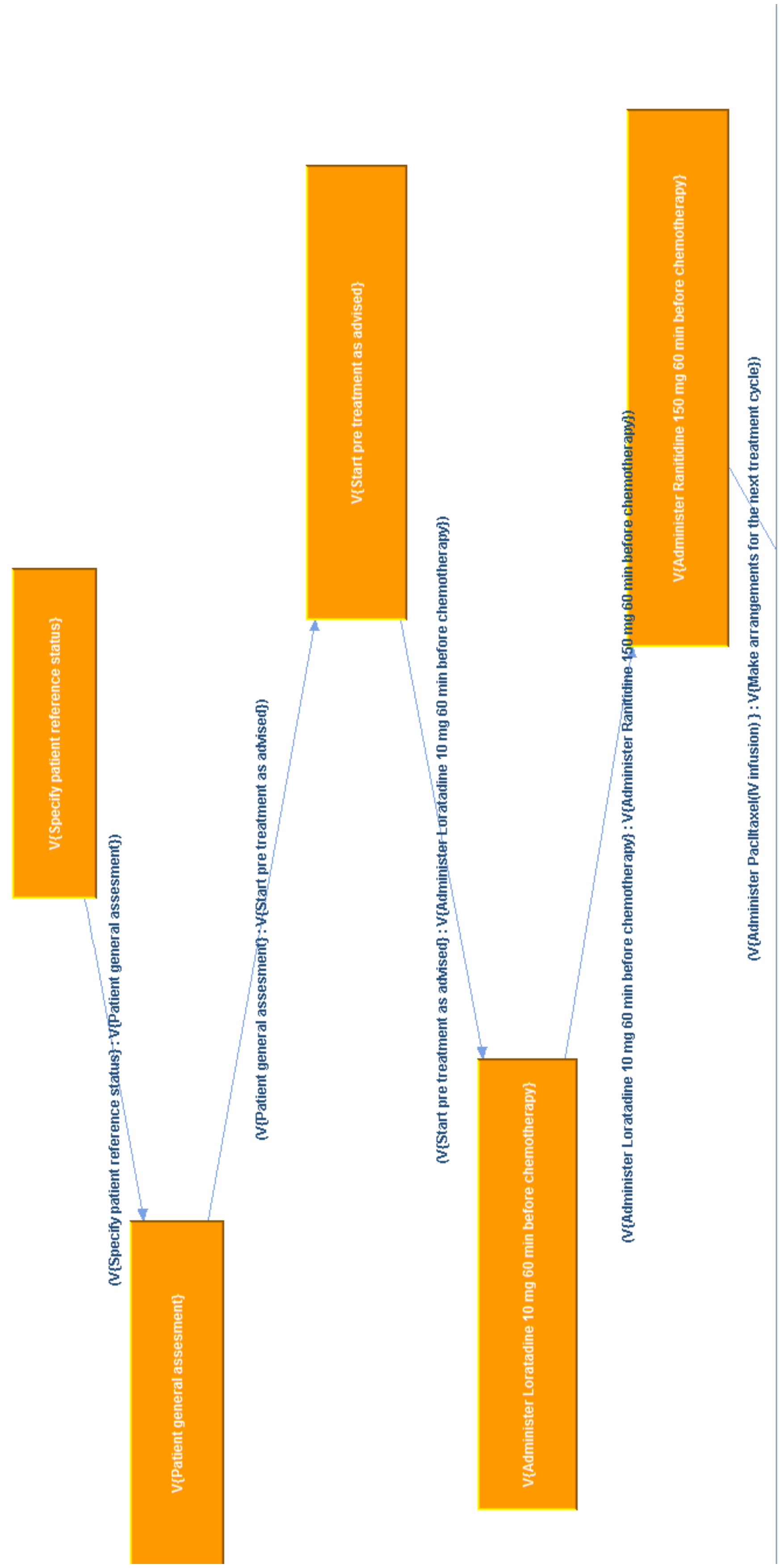

Figure 5.6: Breast cancer protocol treatment CA-Nets- part I

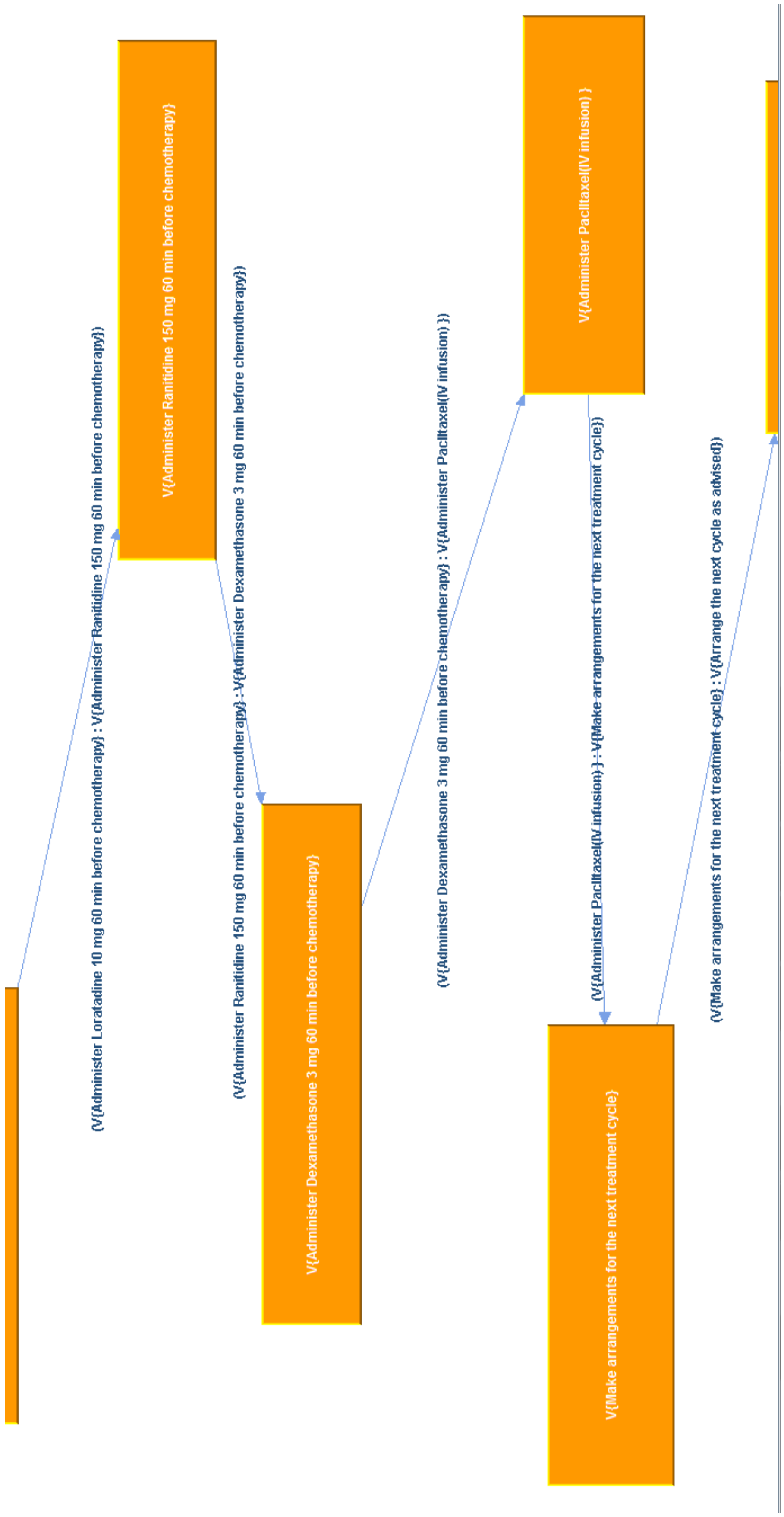

Figure 5.7: Breast cancer protocol treatment CA-Nets-part II

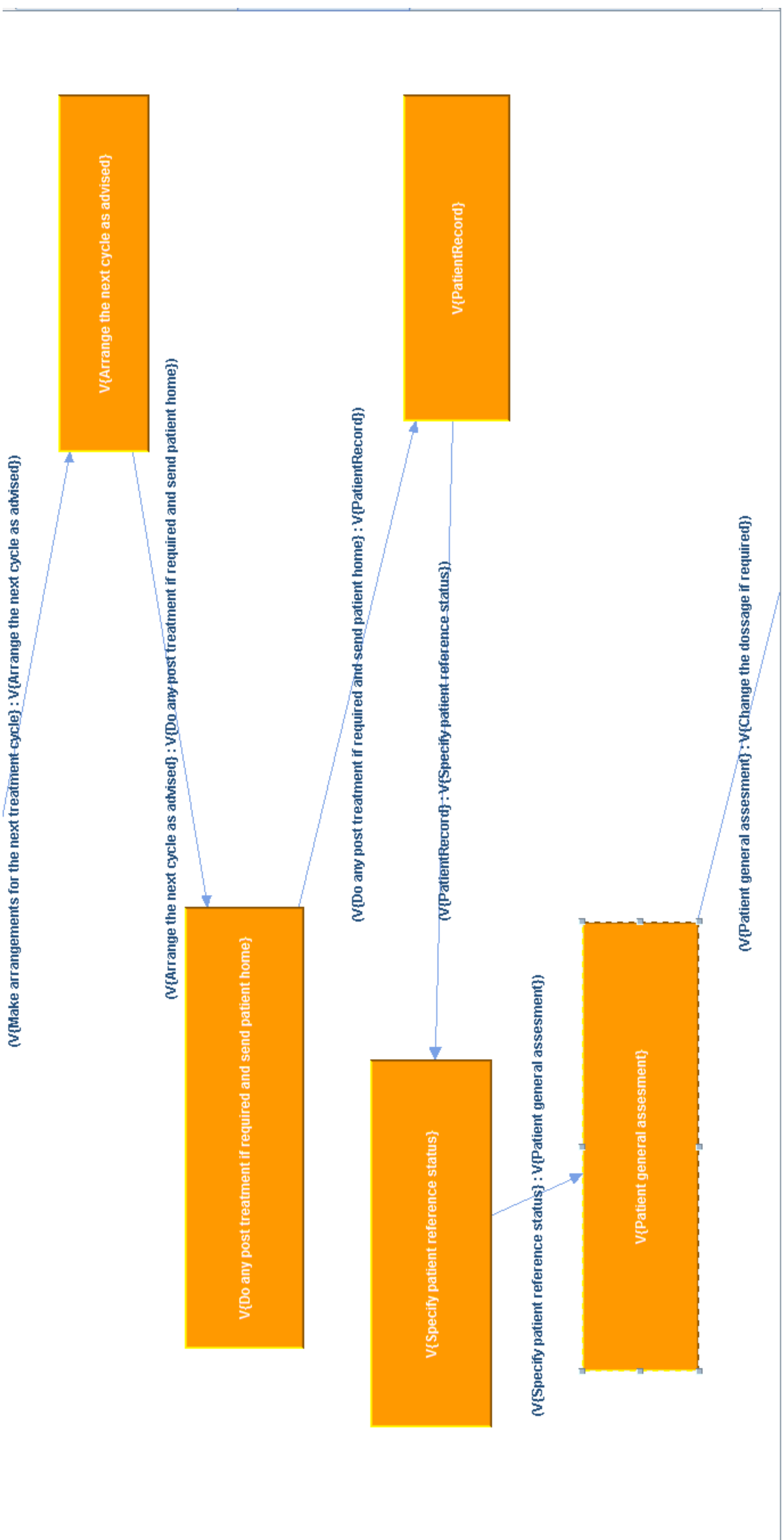

Figure 5.8: Breast cancer protocol treatment CA-Nets-part III

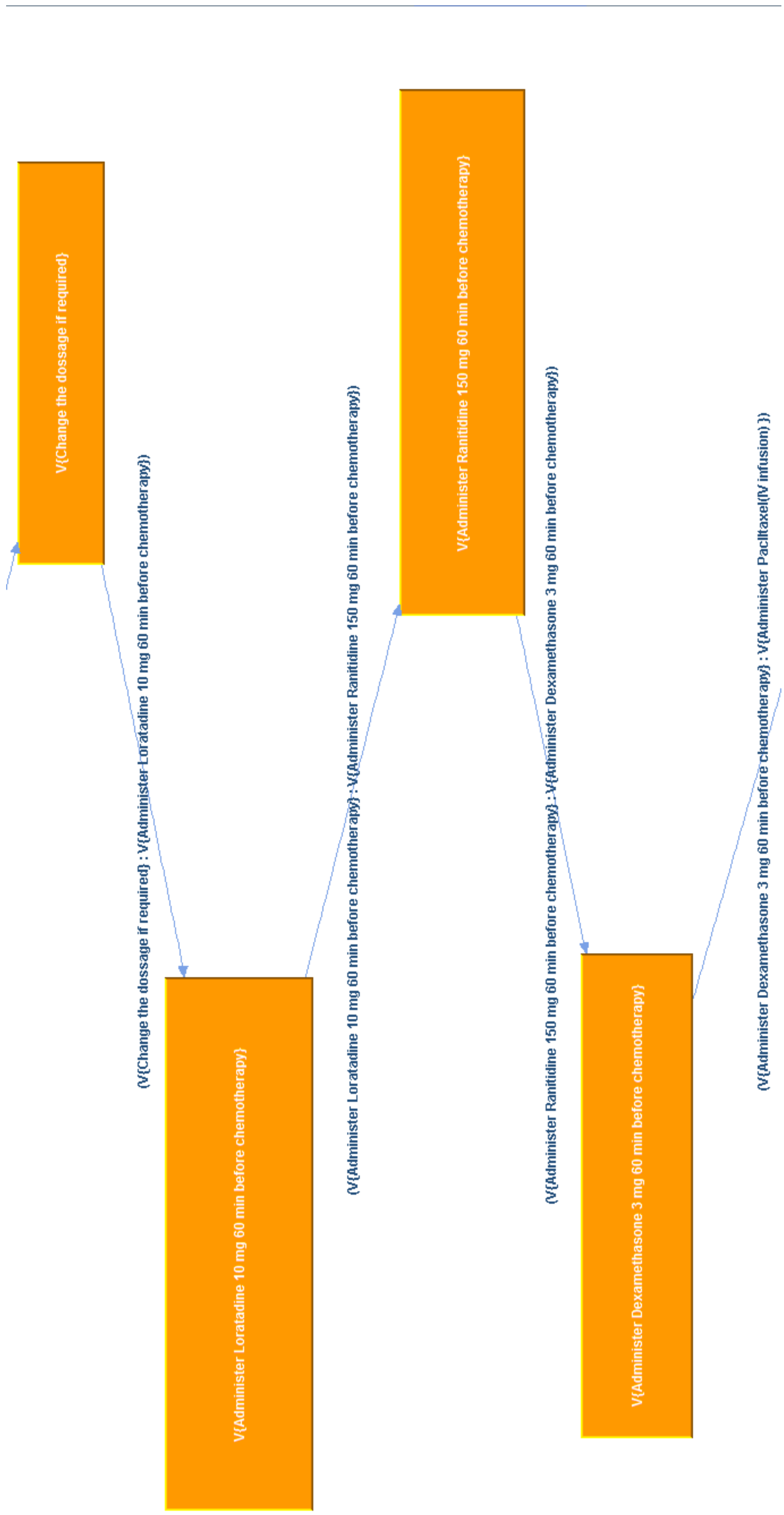

Figure 5.9: Breast cancer protocol treatment CA-Nets-part IV

Figure 5.10: Breast Cancer protocol treatment CA-Nets-part V

5.2.3 Findings

We illustrate the use of CA-Net with two different scenarios. As we can see from the resulting CA-Nets in figures 5.3 and 5.6- 5.10, they illustrate the provenance of the patient history and its evolution in a graphical format. The resulting CA-Nets represent the patient specific treatment for a clinicians in a glance. They support the temporal context queries by tracing back from any node in CA-Nets. It assist clinicians to follow and trace back any nodes in CA-Nets to find about the required information for the patient. It can support answering the queries such as “what is the nearest previous test?” or “what is the nearest following task or document for a diagnosis?”

Moreover, every clinician with the required type of queries can review it and find the artefacts which they are interested to know about. Thus it support different view point for different type of users.

The challenges for supporting other types of queries is the existence of KB to assist finding relation reasons. Thus, current model can not support reasoning and responsibility queries. Implementing and designing a knowledge base as well as using effect annotation for each artefact could be future work to enhance current framework features.

5.3 Conclusion

In this chapter we described the implementation of partial CA-Nets to demonstrate their use with two different examples. We addressed how we can extract the knowledge for a patient specific treatment with two different case scenarios in this chapter. In the first scenario, we illustrated how CA-Nets can be useful in the healthcare system, where a specialist is required to read and review patient health record to find out about the patient’s history. This example is based on our motivating example which was discussed in Chapter 3.

In the second scenario, we demonstrated CA-Nets with a real life example using eviQ

treatment protocols and real provided data via a data log from the local cancer care centre. It shows how CA-Nets can be useful for the clinicians such as doctors and nurses who are in charge of performing the protocol treatment. It assists them to find the deviations and the reasons which caused the deviations for any particular patient treatment in a conceptual visualisation. CA-Nets also help protocol designers and knowledge workers to design and adjust the protocols to make them more accurate. They can redesign the protocols based on the real situations to reduce unnecessary protocol creation. Several protocols in eviQ were analysed and as a result, an abstract process model was proposed which could satisfy all of the protocols. Then one protocol was selected to be analysed in more detail for the purpose of using CA-Nets as a tool for showing patient specific treatment monitoring.

We illustrated the use of our proposed model with two different scenarios in health care systems. This study focused on healthcare systems in particular. Though, a question may raised as whether CA-Nets can be useful in other areas. In response, we claim that the concept of CA-Nets can be applied to any business processes for which there are processes and related documents. CA-Nets can be used to display the data provenance and its evolution when required. In addition, the huge amount of data which is maintained and kept in massive related databases can be retrieved and represented in a contextual and related format to assist and ease the task of knowledge query via the use of CA-Nets.

Chapter 6

Conclusion and Future Work

This chapter presents the key findings of this research and discusses how the research questions have been addressed. The key issue in this study is how to manage the continuously increasing amount of disparate data in health management systems and in particular how to easily retrieve relevant and accurate information from health management systems. Such large quantities of unstructured data, spread across multiple data sources, make it difficult and time consuming for clinicians to locate the information they need to perform their tasks efficiently and accurately. Based on this idea, two questions are addressed in this research which will be discussed in detail in the next section.

The motivation example was designed based on interviews with a radiation oncologist from a local cancer care centre. The problem presented in the motivation example illustrates the need for a way to assist clinicians to view and find the information they need with less effort and time spent. Clinicians need a solution which will assist them to find and retrieve required information from patient health records in relation to patient specific treatments.

The following section reviews and concludes the findings of this research. This is followed by a discussion of the limitations of this study and directions for future research.

6.1 Summary of the Findings

In Chapter 1, two research questions were proposed:

- Given the volume of disparate unstructured data in a range of data stores, how can we extract the required information from these readily available sources?
- How can we represent knowledge about patient history and patient specific treatment in a manner that supports a range of clinical queries?

In order to answer such questions, a graphical conceptual model was proposed. CA-Nets correlate the semantic aspect of the data irrespective of how, where and in what format the data are stored. It is proposed that the resulting model will provide various clinicians and stakeholders the ability to navigate through the collection of data items as one would navigate through a graph or network. With this capability, the model will allow for a range of queries to be answered and also provide the machinery to help justify clinical decisions. To achieve such goals, the model must be generated dynamically as queries to such a network are generally context specific. CA-Nets were proposed to help to support the queries using semantic networks and data provenance techniques.

In Chapter 2, the importance of business process management in healthcare systems was reviewed. This included a literature review of current process management in healthcare systems, and a discussion of the key issues within process management in healthcare systems. In addition, an overview of knowledge representation and context representation was provided. Provenance management was discussed, highlighting the importance of monitoring workflow progress at runtime. In the last section of Chapter 2, the architecture of Process-Oriented Information Logistics (POIL) was discussed and different phases of constructing the Semantic Information Network (SIN), as the core of the POIL architecture, were explained briefly using an example.

Chapter 3 provided a description of how a CA-Net framework can be used, describes CA-Nets framework to represent knowledge for patient specific treatment to support different types of queries. The resulting model will allow clinicians and other decision makers to navigate through the collection of data items that make up a patient history just as one would navigate through a graph or a network. The fundamental machinery of CA-Nets comprises several technology and techniques, with a semantic network at its core. A semantic network was chosen to be the basis of this framework as we found this to be one of the simplest ways of representing knowledge. Moreover, semantic networks could be used to represent patient specific knowledge in a related format to assist different kinds of clinicians with different viewpoints. Data provenance is used to provide complementary metadata prescribing the evolution of data sources and data items. Associated with data to information provenance are machineries to monitor and mine such data evolution, from the original sources of data to information about why data items were created, to processes associated with the data and relationships between data items, through to the current representation of the data.

Furthermore, chapter 3 contained a discussion of the motivations for using CA-Nets, such as obtaining patient context, decision justification, treatment monitoring and the ability to support specific patient viewpoints.

Chapter 4 provided an explanation of how information can be represented for a specific patient treatment to support a range of queries by different clinicians. Building on that foundation, this chapter provides an explanation of how CA-Nets are created from readily available information about patient specific treatment. First, the major elements that make up a CA-Net, such as process log and knowledge base, are defined. Second, the generation of CA-Nets is discussed. Then the types of queries which CA-Nets can support and answer are described, followed by a discussion about supporting reasoning in CA-Nets via the use of effect annotation.

A detailed comparison between the CA-Nets approach and POIL was provided in the last section of chapter 4. It was found that in a SIN every object in the process model such as process, document, relationship, pool, converts into a node in a graph, whereas in CA-Nets only the processes and process related objects convert into artefacts. While this feature makes a SIN a very comprehensive and powerful framework, it is hard to trace the related nodes in the network. Reasoning is supported in both frameworks. In a SIN, the nodes are connected to each other via relation reason (which means the relationship between the two nodes is labelled with the reason behind their connection), but how these relation reasons are created is not discussed.

Chapter 5 illustrated CA-Nets in practice using two scenarios. The steps taken to implement a CA-Net and the architecture of CA-Net were explained. The first scenario was based on the motivation example, where a patient visits a GP. The patient specific context and decision justification was presented together with a demonstration of the sequence of the activities and the documents which are issued as the result of those processes. The second example which is based on a real cancer care scenario and a real data log, displays the treatment monitoring of a particular patient. By accessing the real data for different patients we can answer the request of a Cancer Care Centre clinician regarding which treatment protocols have frequent deviations from the standard treatment protocols as well as determining common reasons behind the deviations.

6.2 Limitations and Future Work

This study proposed a framework to represent knowledge for patient specific treatment to support different types of queries. The resulting model will allow clinicians and other decision makers to navigate through the collection of data items that make up a patient history. However, numbers of limitations are discovered within the resulting model. In cases where the data is incomplete or incorrect, CA-Nets will be an incomplete and not useful output.

The other limitation is that although the real data was used for representing an example of CA-Nets, but the resulting model is not examined or reviewed by real end users. Thus, the usability of CA-Nets is not examined in this study.

In this study the concept of CA-Nets was proposed for an application in healthcare information systems and their use illustrated via two different scenarios. In future work, the use of CA-Nets could be discussed in relation to different scenarios and explore different type of queries that can be supported. This study focused on a small subset of the types of queries which CA-Nets could support so future work could investigate the implementation of CA-Nets which would support other types of queries in other areas of information management and retrieval. This study also proposed the idea of reasoning with CA-Nets. This idea could be discussed in more detail and implemented with the use of a reasoner, thus extending the capability and potential use of CA-Nets.

CA-Nets could be used to solve issues of information management and retrieval in many different fields beyond healthcare. In fact, the idea of CA-Nets can be applied to any business processes for which there are processes and related documents. CA-Nets can be used to display the data provenance and its evolution when required. The huge amount of data which is maintained and kept in massive related databases could be retrieved and represented in a contextual and related format to assist and ease the task of knowledge querying via the use of CA-Nets.

Appendices

Appendix A

CA-Nets code

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.ResultSet;
import java.util.HashSet;
import java.util.LinkedList;
import java.util.TreeMap;
import java.util.List;

import be.fnord.components.dataconnection.myConnection;
import au.edu.dsl.dlab.processtools.Edge;
import au.edu.dsl.dlab.processtools.Effect;
import au.edu.dsl.dlab.processtools.Graph;
import au.edu.dsl.dlab.processtools.Vertex;
import au.edu.dsl.dlab.processtools.parser.bpmn.visualize;
import au.edu.dsl.dlab.processtools.scenario.ComputeScenario;
```

```
public class CANetGenerator {

public static TreeMap<String, String> patientID =
new TreeMap<String, String>();

// PaitnetID, Patient Name

public static TreeMap<String, LinkedList<String>> patientCase =
new TreeMap<String, LinkedList<String>>();

// PatientID, patient task history

public static int counter = 0;
public static void main(String [] args){
 myConnection newInstance = new myConnection();
 boolean test1 = newInstance.makeConnection(``activiti",``root",``");

// Get Patients / Process References

ResultSet rs =
newInstance.executeQuery
(``SELECT * FROM ACT_HI_VARINST WHERE NAME_ like 'patientName';");

LinkedList<String[]> dat =
```

```
(LinkedList<String[]>) newInstance.getMetaDataResults(rs);
TreeMap<String, List<Object>> das =
newInstance.getDataResults(rs);
TreeMap<String, String> patientReference =
new TreeMap<String, String>();
for(String key:das.keySet()){
        int i = 0;
        String patientName = ‘‘”;
        String processReference = ‘‘”;
        for(Object val : das.get(key)){
         if(++i == 2) processReference = val.toString();
         if(i == 11) patientName = val.toString();
                 }
patientReference.put(processReference, patientName);
            }
System.out.println(”tes” + patientReference);

// for each process instance check if it is a new

ResultSet rst =newInstance.executeQuery
(”SELECT * FROM ACT_HI_VARINST WHERE NAME_ like 'referanceStatus';”);
 LinkedList<String[]> datt =
(LinkedList<String[]>) newInstance.getMetaDataResults(rst);
TreeMap<String, List<Object>> dast =
newInstance.getDataResults(rst);
TreeMap<String, String> patientReferencet =
```

```
new TreeMap<String , String >();
for(String keyt:dast.keySet()){
int i = 0;
String input = "";
String processReferencet = "";
for(Object valt : dast.get(keyt)){
if(++i == 2) processReferencet = valt.toString();
if(i == 11) input = valt.toString();
                    }
patientReferencet.put(processReferencet, input);
            }
System.out.println("tes" + patientReferencet);

// For each patient, get any associated process models.

// Get process from database

for(String procID : patientReference.keySet()){

String name = patientReference.get(procID) ;
String inputname = patientReferencet.get(procID);
System.out.println( a.e.endl + "Processes for patient " +inputname);
String sql = "SELECT * FROM ACT_HI_ACTINST WHERE PROC_INST_ID_ =
'"+procID+"'and TASK_ID_ <> 'NULL' ORDER BY ID_ ASC;";
rs = newInstance.executeQuery(sql);
dat = (LinkedList<String[]>) newInstance.getMetaDataResults(rs);
```

```
das = newInstance.getDataResults(rs);
String processName = "";
LinkedList<String> tasks = new LinkedList<String>();
for(String key:das.keySet()){
int i = 0;

for(Object val : das.get(key)){
if(val == null) continue;
if(++i == 7) tasks.add(val.toString());
if(i == 2) processName = val.toString();
                                }
                            }

// Get related document attachment to each process instance

String sql1 = "SELECT * FROM act_hi_attachment WHERE PROC_INST_ID_='"
rst = newInstance.executeQuery(sql1);
datt = (LinkedList<String[]>) newInstance.getMetaDataResults(rst);
dast = newInstance.getDataResults(rst);
for(String keyt:dast.keySet()){
        int j = 0;
        for(Object valt : dast.get(keyt)){

        if (valt == null)continue;
        if(++j == 3) tasks.add(valt.toString());
                }
```

```
                }

System.out.println(tasks.toString());
System.out.println("Process is " + processName);

// if the patient name is in the system check if
//the current process is related to the previous
//process instance for a particular patient

if(patientID.values().contains(name)){
        System.out.println("Process is " + inputname);

        String inputstring = "followup";
if (inputname != null){
        if (inputname.equals(inputstring)){

System.out.println("Adding to your patient history.");
String pID = "";
for(String s: patientID.keySet()){
if(patientID.get(s).compareTo(name) == 0) pID = s;
LinkedList<String> history = patientCase.get(pID);
history.addAll(tasks);
patientCase.remove(pID);
patientCase.put(pID, history);
System.out.println("Updated case history for
" + name + " is " + history);
```

```
                                }
                            }
else{
System.out.println("Okay.");

// Store this patient record in a treemap

String ID = name+'#'+counter;
patientID.put(ID, name);
patientCase.put(ID, tasks);
                             }
                    }

//if it is not related to the previous existing
process instances create a new CA-Nets

else{

// Store this patient record in a treemap

        String ID = name+'#'+counter;
        patientID.put(ID, name);
        patientCase.put(ID, tasks);
                    }
            }
```

```
          //create the CA–Nets

for(LinkedList<String> t : patientCase.values()){
           Graph<Vertex, Edge> myProcess = new Graph<Vertex, Edge>();
           Vertex prev = null;
          for(String task: t){
          Vertex t1 = new Vertex(task); t1.addEffect("aa");
          myProcess.addVertex(t1);
          if(prev != null){
          myProcess.addEdge(prev,t1);
          //myProcess.addEdge(t1,prev);
          prev = t1;
                    };
prev = t1;
              }
visualize<Vertex,Edge> myViewer = new visualize<Vertex,Edge>();
myViewer.showModel(myProcess);


                    }
HashSet<Effect> finalEffects ;


boolean test4 = (newInstance.closeConnection());
          }


}
// The following part is the class which generates the graph
```

```
package au.edu.dsl.dlab.processtools.parser.bpmn;

import java.util.HashMap;
import java.util.Map;

import javax.swing.BoxLayout;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JScrollPane;

import org.jgraph.JGraph;
import org.jgraph.graph.DefaultGraphCell;
import org.jgraph.graph.GraphConstants;
import org.jgraph.graph.AttributeMap.SerializableRectangle2D;
import org.jgrapht.ListenableGraph;
import org.jgrapht.ext.JGraphModelAdapter;
import org.jgrapht.graph.ListenableDirectedGraph;

import au.edu.dsl.dlab.processtools.Edge;
import au.edu.dsl.dlab.processtools.Graph;
import au.edu.dsl.dlab.processtools.Vertex;


public class visualize<T extends Vertex, V extends Edge> {
public transient JGraphModelAdapter<T,V> m_jgAdapter;
```

```
public void showModel(Graph<T,V> output){
ListenableGraph<T,V> g = new ListenableDirectedGraph<T,V>(output );
m_jgAdapter = new JGraphModelAdapter<T,V>( g );
JGraph jgraph = new JGraph( m_jgAdapter );
for(T vert : output.vertexSet()){
positionVertexAt(vert, vert.x, vert.y);
            }
 JFrame frame = new JFrame();
        BoxLayout boxLayout =
        new BoxLayout(frame.getContentPane(), BoxLayout.Y_AXIS);
  frame.setLayout(boxLayout);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.add(new JLabel("Effect Scenarios: " + output.effects));
  frame.add(new JScrollPane(jgraph));
  frame.pack();
        frame.setSize(frame.getWidth(), frame.getHeight() + 100);
        frame.setVisible(true);

        }

private void positionVertexAt( T vertex, int x, int y ) {
 DefaultGraphCell cell = m_jgAdapter.getVertexCell( vertex );
 Map<?, ?> attr = cell.getAttributes( );
 SerializableRectangle2z b =
(SerializableRectangle2D) GraphConstants.getBounds( attr );
```

```
GraphConstants.setBounds(attr, new SerializableRectangle2D
( Math.pow(x,1.1), Math.pow(y,1.1), b.width, b.height ) );
Map<DefaultGraphCell,Map<?, ?>>cellAttr=
new HashMap<DefaultGraphCell, Map<?, ?>>( );
cellAttr.put( cell, attr );
m_jgAdapter.edit( cellAttr, null, null, null );
      }

}
```

Bibliography

[1] M. Alberti, M. Gavanelli, E. Lamma, P. Mello, and P. Torroni. Abduction with hypotheses confirmation. In *of the 19th international joint conference on Artificial intelligence*, pages 1545–1546, 2005.

[2] K. Anyanwu, A. Sheth, J. Cardoso, J. Miller, and K. Kochut. Healthcare enterprise process development and integration. *Journal of Research and Practice in Information Technology*, 35(2):83–98, 2003.

[3] T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web. *Scientific American*, 284(5):34–43, 2001.

[4] I. A. Bessmertny. Knowledge visualization based on semantic networks. *Programming and Computer Software*, 36(4):197–204, 2010.

[5] M. Born, J. Hoffmann, T. Kaczmarek, M. Kowalkiewicz, I. Markovic, J. Scicluna, and Zhou X Weber, I. Semantic annotation and composition of business processes with maestro. *European Semantic Web Conference (ESWC) Demo Track*, 2008.

[6] E.M. Borycki, A.W. Kushniruk, S. Kuwata, and J. Kannry. Use of simulation approaches in the study of clinician workflow. In *AMIA Annual Symposium Proceedings. 61-5*, 2006.

[7] P. Buneman, S. Khanna, and W. C. Tan. Why and where: A characterization of data provenance. *ICDT*, pages 316–330, 2001.

[8] T. Burkle, T. Baur, and N Hoss. Clinical pathways development and computer support in the epr: Lessons learned. *Studies in Health Technology and Informatics*, 124:1025–1030, 2006.

[9] F. Caron, J. Vanthienen, J. Weerdt, and B. Baesens. Beyond x-raying a care-flow: Adopting different focuses on care-flow mining. 2011.

[10] F. Chesani, E. Lamma, P. Mello, M. Montali, S. Storari, P. Baldazzi, and M. Manfredi. Compliance checking of cancer-screening careflows: An approach based on computational logic. In *Studies in Health Technology and Informatics*, volume 139, pages 183–192, 2008.

[11] F. Chesani, P. Mello, M. Montali, and S. Storari. Testing careflow process execution conformance by translating a graphical language to computational logic. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4594 LNAI, pages 479–488, 2007.

[12] C. M. Chiao, V. Künzle, and M. Reichert. Towards object-aware process support in healthcare information systems. In *4th International Conference on eHealth, Telemedicine, and Social Medicine (eTELEMED 2012)*, pages 227–236. IARIA, February 2012.

[13] M. Chinosi and A. Trombetta. BPMN: an introduction to the standard. *Computer Standards and Interfaces*, 34(1):124–134, 2012.

[14] T. Connolly and J. Reb. Regret in cancer-related decisions. *Health Psychology*, 24(4 SUPPL.):S29–S34, 2005.

[15] R. Curia, L. Gallucci, and M. Ruffolo. Knowledge management in health care: An architectural framework for clinical process management systems. In *Proceedings*

of the International Workshop on Database and Expert Systems Applications, DEXA, pages 393–397, 2006.

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

[17] J. Dallien, W. MacCaull, and A. Tien. Initial work in the design and development of verifiable workflow management systems and some applications to health care. In *Proceedings of the 5th International Workshop on Model-Based Methodologies for Pervasive and Embedded Software, MOMPES 2008*, pages 78–91, 2008.

[18] S. B. Davidson and J. Freire. Provenance and scientific workflows: challenges and opportunities. In *In Proceedings of ACM SIGMOD*, pages 1345–1350, 2008.

[19] M. Denecker and A. C. Kakas. Abduction in logic programming,. *Computational Logic: Logic Programming and Beyond*, pages 402–436, 2002.

[20] D. Gannon, B. Plale, M. Christie, L. Fang, Y. Huang, S. Jensen, G. Kandaswamy, S. Marru, S.L. Pallickara, S. Shirasuna, Y. Simmhan, A. Slominski, and Y. Sun. Service oriented architectures for science gateways on grid systems. In *Proceedings of the Third International Conference on Service-Oriented Computing*, 2005.

[21] E. Gattnar, O. Ekinci, and V. Detschew. Clinical process modeling and performance measurement in hospitals. In *Proceedings of the IEEE International Enterprise Distributed Object Computing Conference, EDOC*, pages 132–140, 2011.

[22] V. Geol and R.J. Dolan. Differential involvement of left prefrontal cortex in inductive and deductive reasoning. *Cognition*, B:109–121, 1993.

[23] A. Ghose and G. Koliadis. *Auditing business process compliance*, volume 4749 LNCS of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics).* 2007.

[24] M. Greenwood, C. Goble, R. Stevens, J. Zhao, M. Addis, D. Marvin, L. Moreau, and T. Oinn. Provenance of e-science experiments - experience from bioinformatics. In *UK OST e-Science second All Hands Meeting*, 2003.

[25] C. W. Günther, M. Reichert, and van der Aalst. Supporting flexible processes with adaptive workflow and case handling. In *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WET ICE*, pages 229–234, 2008.

[26] A. R. Hani Tawil, A. Litnhouvongs, Q. Chevalier, and A. Taweel. An ontology-based approach to adaptive home careflows. *International Workshop on Database and Expert Systems Applications*, 1:522–526, 2011.

[27] R. T. Hartley and J. A. Barnden. Semantic networks: Visualizations of knowledge. *Trends in Cognitive Sciences*, 1(5):169–175, 1997.

[28] K. Hinge, A. Ghose, and A. Miller. A framework for detecting interactions between co-incident clinical processes. *Journal of E-Health and Medical Communications*, 1(2):24–35, 2010.

[29] M. Hipp, B. Michelberger, B. Mutschler, and M. Reichert. A framework for the intelligent delivery and user-adequate visualization of process information. In *in:28th Symposium on Applied Computing (SAC 2013), Coimbra, Portugal. ACM Press (March 2013)*, 2013.

[30] M. Hipp, B. Mutschler, and M. Reichert. On the context-aware, personalized delivery of process information: Viewpoints, problems, and requirements. In *Proceedings of*

the 6th International Conference on Availability, Reliability and Security, ARES 2011, pages 390–397, 2011.

[31] D. Hollingsworth. Workflow management coalition: Workflow reference model. Technical Report TLOO-1003, Workflow Management Coalition, 1994.

[32] B. Huang, P. Zhu, and C. Wu. Customer-centered careflow modeling based on guidelines. *Journal of Medical Systems*, 1:1–13, 2012.

[33] P.N. Johnson Laird. Deductive reasoning. *Annual review of psychol*, 50:109–135, 1999.

[34] J.R. Josephson and S. G. Josephson. *Abductive inference: Computation, philosophy, technology*. Cambridge, England: Cambridge University Press., 1994.

[35] B. Kiepuszewski, A. H. M. ter Hofstede, and W. M. P. van der Aalst. Fundamentals of control flow in workflows. *Acta Informatica*, 39(3):143–209, 2003.

[36] I. Kulev, E. Janevska, M. Jovanovik, R. Stojanov, and D. Trajanov. Text classification using semantic networks. In *Proceedings of the 8th International Conference for Informatics and Information Technology*, 2011.

[37] M. Lang, T. Bürkle, S. Laumann, and H. Prokosch. Process mining for clinical workflows: Challenges and current limitations. In *in Proceeding of MIE Studies in Health Technology and Informatics*, volume 136, pages 229–234, 2008.

[38] R. Lenz, M. Peleg, and M. Reichert. Healthcare process support: Achievements, challenges, current research. *International Journal of Knowledge-Based Organizations (IJKBO)*, 2(4), 2012.

[39] J. Leu and Y. Huang. An application of business process method to the clinical efficiency of hospital. *Journal of Medical Systems*, 35:409–421, 2011.

[40] R.S. Mans, H. Schonenberg, G. Leonardi, S. Panzarasa, A. Cavallini, S. Quaglini, and W.M.P. van der Aalst. Process mining techniques: An application to stroke care. In *Studies in Health Technology and Informatics*, volume 136, pages 573–578, 2008.

[41] L. Maruster, W. van der Aalst, T. Weijters, A. van der Bosch, and W. Daelemans. Automated discovery of workflow models from hospital data. *Proceedings of the13th Belgium-Netherlands Conference on Artificial Intelligence*, 1:183–190, 2001.

[42] K. Maximini and M. Shaaf. The PROGEMM approach for managing clinical processes. In *Proceedings of the twelfth IEEE International workshops on enabling technologies: Infrastructure for collaborative enterprises(WETICE03), Washington, DC:IEEE*, pages 332–337, 2003.

[43] B. Michelberger, B. Mutschler, and M. Reichert. A context framework for process-oriented information logistics. In *In Proc.15th Int'l Conference on Business Information Systems (BIS 2012)*, LNBIP, pages 260–271. Springer, 2012.

[44] B. Michelberger, B. Mutschler, and M. Reichert. Process-oriented information logistics: Aligning enterprise information with business processes. In *In Proc. of the16th IEEE International EDOC Conference (EDOC 2012)*, pages 21–30. IEEE Computer Society Press, September 2012.

[45] B. Mikolajczak and M. Shenoy. Flexibility through case handling in careflow systems: A case study of cutaneous melanoma. In *IEEE Workshop on Health Care Management, WHCM*, 2010.

[46] K. Miller and W. MacCaull. Toward web-based careflow management systems. *Journal of Emerging Technologies in Web Intelligence (JETWI)*, 1(2):137–145, 2009.

[47] R. Müller, U. Greiner, and E. Rahm. Agentwork: A workflow system supporting rule-based workflow adaptation. *Data and Knowledge Engineering*, 51:223–256, 2004.

[48] J. Munoz-Gama and J. Carmona. A fresh look at precision in process conformance. In *in Proceedings of the 8th International Conference on Business Process Management*, 2010.

[49] S. Najar, O. Saidani, M. Kirsch-Pinheiro, C. Souveyet, and S. Nurcan. Semantic represerepresе of context models: a framework for analyzing and understanding. In *the 1st Workshop on Context, Information and Ontologies (CIAO 09)*. European Semantic Web Conference (ESWC'2009), ACM, 1-10, 2009.

[50] H. Niemann, G. F. Sagerer, S. Schroder, and F. Kummert. ERNEST: A semantic network system for pattern understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(9):883–905, 1990.

[51] S. Panzarasa, R. Bellazzi, C. Larizza, and M. Stefanelli. A careflow management system for chronic patients. *Medinfo*, 11:673–677, 2004.

[52] S. Panzarasa, S. Madd, S. Quaglini, C. Pistarini, and M. Stefanelli. Evidence-based careflow management systems: The case of post-stroke rehabilitation. *Journal of Biomedical Informatics*, 35(2):123–129, 2002.

[53] S. Panzarasa and M. Stefanelli. Workflow management systems for guideline implementation. *Neurological Sciences*, 27(SUPPL. 3):S245–S249, 2006.

[54] M. Peleg, P. Soffer, and J. Ghattas. Mining process execution and outcomes - position paper. 4928 LNCS:395–400, 2008.

[55] S. Peters and H. E. Shrobe. Using semantic networks for knowledge representation in an intelligent environment. In *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications, PerCom 2003*, pages 323–329, 2003.

[56] H. A. Reijers, N. Russell, S. Van Der Geer, and G. A. M. Krekels. *Workflow for healthcare: A methodology for realizing flexible medical treatment processes*, volume

43 LNBIP, pp 593-604 of *Lecture Notes in Business Information Processing*. Springler, 2010.

[57] S. Rinderle, M. Reichert, and P. Dadam. Correctness criteria for dynamic changes in workflow systems - a survey. *Data and Knowledge Engineering*, 50:9–34, 2004.

[58] A. Rozinat and W. M. P. van der Aalst. Conformance checking of processes based on monitoring real behavior. *Information Systems*, 33(1):64–95, 2008.

[59] M. Ruffolo, M. Manna, V. Cozza, and R. Ursino. Semantic clinical process management. In *in CBMS, in Proceedings of the IEEE Symposium on Computer-Based Medical Systems*, pages 518–523, 2007.

[60] K. Sato. Context sensitive interactive systems design: A framework for representation of contexts. In *the 10th International Conference on Human Computer Interaction*, 2003.

[61] H. Scheuerlein, F. Rauchfuss, Y. Dittmar, R. Molle, T. Lehmann, N. Pienkos, and U. Settmacher. New methods for clinical pathways-business process modeling notation (BPMN) and tangible business process modeling (t.BPM). *Langenbeck's Archives of Surgery*, 397:775–761, 2012.

[62] K. Shah and B. Mikolajczak. Careflow system with case-handling flexibility - case study of gastric cancer. In *Proceedings of the ITNG2010 - 7th International Conference on Information Technology: New Generations*, 2010.

[63] Y. L. Simmhan, B. Plale, and D. Gannon. A survey of data provenance techniques. Technical report tr-618:, Computer Science Department, Indiana University, 2005.

[64] Y. L. Simmhan, B. Plale, and D. Gannon. A framework for collecting provenance in data-centric scientific workflows. *Proceedings of the International Conference on Web Service (ICWS)*, 2006.

[65] F. Sowa John. *Principles of semantic networks: expolrations in the representation of knowledge*. Morgan Kaufmann, San Mateo, CA, 1991.

[66] T. Strang and C. Linnhoff-Popien. A context modelling survey. In *In Proceedings of the Workshop on Advanced Context Modelling, Reasoning and Management within UbiComp04, 1-8.*, volume 1, 2004.

[67] J. Sutherland and W. Van Den Heuvel. Towards an intelligent hospital environment: Adaptive workflow in the OR of the future. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, volume 5, 2006.

[68] L. Thom, M. Reichert, C. Iochpe, and J.P. M. de Oliveira. Why rigid process management technology hampers computerized support of healthcare processes? In *X Workshop on Medical Informatics (WIM'2010), Workshop held in conjunction with SBC'10 Congress*, pages 1522–1531, 2010.

[69] W.M.P. van d e r Aa l s t. Petri-net -based workflow management software. In A. Sheth, editor, *Proceedings of the NFS Workshop on Workflow and Process Automation in Information Systems*, 1996.

[70] W. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.

[71] W. M. P. van der Aalst. The application of petri nets to workflow management. *Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998.

[72] W. M. P. van der Aalst and M. Pesic. Decserflow: Towards a truly declarative service flow language. In *In Proceedings of the 3rd International Workshop on Web Services and Formal Models*, volume 4184 LNCS of *Lecture Notes in Computer Sci-*

ence (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pages 1–23, 2006.

[73] W. M. P. van der Aalst and M. Pesic. Specifying and monitoring service flows: Making web services process-aware. *Test and Analysis of Web Services*, 2007.

[74] W. M. P. van der Aalst and A. H. M. Ter Hofstede. YAWL: yet another workflow language. *Information Systems*, 30(4):245–275, 2005.

[75] W. M. P. van der Aalst, A. H. M. Ter Hofstede, and M. Weske. *Business process management: A survey*, volume 2678 of LNCS, pages 1-12 of *Lecture Notes in Computer Science*. Springer, 2003.

[76] W. M. P. van der Aalst, B. F. Van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. J. M. M. Weijters. Workflow mining: A survey of issues and approaches. *Data and Knowledge Engineering*, 47(2):237–267, 2003.

[77] W.M.P. van der Aalst. Formalization and verification of event-driven process chains. *Information and Software Technology*, 41(10):639–650, 1999.

[78] B. F. Van Dongen, A. K. Alves de Medeiros, and L. Wen. *Process mining: Overview and outlook of Petri net discovery algorithms*, volume 5460 LNCS, pages 225-242 of *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer-Verlag, 2009.

[79] K. Vanhaecht, K. De Witte, R. Depreitere, and W. Sermeus. Clinical pathway audit tools: A systematic review. *Journal of Nursing Management*, 14(7):529–537, 2006.

[80] K. Verlaenen, W. Joosen, and P. Verbaeten. Arriclides: An architecture integrating clinical decision support models. In *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2007.

[81] D. Wang, J. Ge, H. Hu, B. Luo, and L. Huang. Discovering process models from event multiset. *Expert Systems with Applications*, 2012. Article in Press.

[82] M. Weske, W. M. P. van der Aalst, and H. M. W. Verbeek. Advances in business process management. *Data and Knowledge Engineering*, 50(1):1–8, 2004.

[83] D. B. West. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.