THE UNIVERSITY OF
SYDNEY

THE UNIVERSITY OF
SYDNEY

An Extended 4D Fluent Analysis of Temporal Knowledge in OWL-Based Clinical Guideline System

Hongsheng Li

A thesis submitted in fulfilment of the requirements for the degree of Master of Philosophy

School of Information Technologies
University of Sydney

01 September 2014

Abstract

The Web Ontology Language (OWL) based clinical guideline system is a kind of clinical decision support system which is often used to assist health professionals to find clinical recommendations from the guidelines and check clinical compliance issues in terms of the guideline recommendations. However, due to some limitations of the current OWL language constructs, temporal knowledge contained in various knowledge domains cannot be directly represented in OWL. As a result, the representation, query and reasoning of temporal knowledge are largely ignored in many OWL-based clinical guideline ontology systems.

The aim of this research is to investigate a temporal knowledge modelling method namely "4D fluent" and extend it to represent the temporal constraints contained in clinical guideline recommendations within OWL language constructs. The extended 4D fluent method can model temporal constraints including valid calendar time, interval, duration, repetitive or cyclical temporal constraints and temporal relations such that it can enable reasoning over these temporal constraints in the OWL-based clinical guideline ontology system and overcome the shortcoming of the traditional OWL-based clinical guideline system to an extent.

A prototype clinical guideline ontology system is built from the "Intensive Care Unit Empirical Antimicrobial Treatment Guidelines" written by QUAIC (Quality Use of Antimicrobials in the Intensive Care Unit) expert group for local NSW hospitals to demonstrate the extended 4D fluent method. The prototype system also leverages the international standard medical terminology SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms) to organise the medical concepts in that guideline such that it can facilitate the medical terminology interoperability.

Acknowledgements

Foremost, I would like to express my sincere thankfulness to my supervisor Dr Josiah Poon, who gave me the timely guidance, insightful comments and encouragement for this challenging research project. This project would not have been possible without his generous and continuous support.

Besides my supervisor, my sincere thanks also goes to my former supervisor Prof. Jon Patrick, the medical language research group members Ying Ou, Zhizhou Yin, Ming Li, Leila Safari and Pooyan Asgari for their good advice and kindly help throughout this project.

I am most grateful to Dr Stephen Crawshaw, who helped me with his immense medical expertise for interpreting the medical knowledge in the clinical guidelines and database, and verifying the medical knowledge and results in the prototype guideline ontology system. Without his valuable advice, the project would not have turned out the way I expected.

Last, but by no means least, I would like to send my appreciation to my whole family. Words cannot express how grateful I am for their solid backing throughout my research period.

Table of Contents

Chapter 1 Introduction

1.1 Background

A clinical practice guideline is an important type of free text clinical document in health care institutions which contains "systematically developed statements to assist practitioner and patient decisions about appropriate health care for specific clinical circumstances" (Field & Lohr 1990. page 38) [1]. It is a document containing "recommendations and instructions to assist the medical professional and the patient in decision making, based on results of scientific research followed by discussion and expression of expert-opinions, to make effective and efficient medical practice explicit" (The Dutch Institute for Healthcare Improvement CBO. page 5) [2]. Clinical guidelines play an important role in improving the health care quality of actual clinical practice.

In order to facilitate the acceptance and application of clinical guidelines in daily health care, many research groups from both the information technology and health care industries are developing computerised clinical guideline systems, computer based clinical guidelines, or computer interpretable guidelines such that they can be used as a clinical decision support system to assist clinicians to find treatment recommendations for their patients and checking medical compliance issues with these recommendations. The outcomes will help clinicians review and research their clinical practice with regard to the guidelines.

Many models and formal languages have been proposed in the research field of clinical guideline representation, query and reasoning. OWL-based formalism is one of the important approaches for knowledge representation and reasoning in computerised clinical guideline systems. OWL, which is developed in the Semantic Web research field, provides a formalised vocabulary to describe concepts in the domain and relationship between these concepts. The recent development of OWL is OWL 2. The OWL language constructs such as class, individual, object property, data property, property characteristic, property restriction, cardinality restriction and property chain inclusion make it possible to precisely describe the knowledge in the domain of interest so that the computer can interpret and manipulate it in programs such as knowledge reasoning. Many tools have been developed to support OWL

ontology creation, visualization, query and reasoning. OWL has been widely used in different knowledge domains. Medical science is one of the important areas where OWL has achieved success. For example, the most comprehensive and largest clinical terminology SNOMED CT is an important application of OWL which is modelled in OWL 2 EL, a less expressive language in the OWL 2 Profile family. OWL-based computerised clinical guideline systems have also gained a lot of research interest in the medical domain. There are already some important applications in this area such as diagnosis and clinical management of patients with diabetic retinopathy disease, anti-diabetic drug recommendation, contraindication and side effect monitoring for diabetic patients, patient specific recommendation of follow up care for breast cancer patients, and treatment recommendation for patients in the cardiac intensive care units.

1.2 Research Problem and Motivation

OWL is a knowledge modelling language which is based on a binary predicate and does not directly support the representation of knowledge which is based on predicates with higher arity such as ternary predicate. Due to the limitation of the current OWL standard, some important knowledge types cannot be modelled directly in OWL. Subsequently, the related query and reasoning tasks cannot be implemented directly in OWL reasoners. For example, OWL does not support knowledge containing uncertainty, propositional attitudes, epistemic and deontic modalities. Knowledge containing temporal constraints is also an important one which cannot be represented directly by OWL.

Temporal knowledge is an essential and indispensable part of various knowledge domains including medical domains such as clinical guidelines. Knowledge in different domains often contains different temporal constraints. The most common ones are valid calendar time point and interval constraints on events or activities. Another important one is the temporal relation constraint such as the ordering or sequence of events. Moreover, temporal constraints in different knowledge domains often involve repetition in a certain temporal pattern, relativity, duration, indeterminacy, delay and fuzziness. Especially in clinical guidelines, constraints involving temporal relation, repetition, relativity, duration, indeterminacy, delay and fuzziness are very common.

Due to the limitation of OWL, the representation, query and reasoning of temporal knowledge is largely ignored in many OWL-based clinical guideline ontology systems. This prevents the application of OWL in the computerised clinical guideline system.

1.3 Research Contribution

Motivated by the issue of temporal knowledge representation, query and reasoning in OWL-based clinical guideline systems, this research project investigates a temporal knowledge modelling method, namely 4D fluent, to represent some important temporal constraints contained in the recommendations of clinical guidelines. The main contribution of this research project is that it presents an extended 4D fluent method which can be used to represent and reason with temporal constraints of clinical guidelines in the OWL-based clinical guideline system. The extended 4D fluent method does not modify the underpinning Description Logic of OWL but rather works at the user level within OWL language constructs to represent these temporal constraints.

In this extended 4D fluent method, the temporal ontology proposed in the original 4D fluent method is extended such that it can model temporal constraints including valid calendar time, interval, duration, repetitive or cyclical temporal constraints and temporal relations in the OWL-based clinical guideline systems. For the repetitive temporal constraint, this extended temporal ontology can be used to compute the length of a time interval between two adjacent events in a time series such as dose interval of antibiotic administration and the length of the time period that an event lasts from start to end such as dose duration of antibiotic administration. This type of temporal constraint is particularly important for clinical practice compliance checking with regard to the guidelines such as drug administration compliance checking where fixed periodical intervals between doses need to be followed for safety and efficacy purposes. To the best of our knowledge, this type of temporal constraint has not been investigated in the current literatures of 4D fluent temporal knowledge representation and reasoning. For the temporal relations between clinical activities or events, 13 Allen's basic temporal relations and 14 Allen's indefinite fuzzy temporal relations are modelled in the ontology. These 27 relations are used in temporal relation reasoning based on the Constraint Propagation Algorithm of Allen's interval algebra for finding the exact temporal relations between clinical events and checking the inconsistent temporal relations which might occur in the ontology.

The extended 4D fluent method is demonstrated in a prototype of OWL-based antibiotic treatment guideline ontology system which is derived from the "Intensive Care Unit

Empirical Antimicrobial Treatment Guidelines" written by QUAIC expert group for local NSW hospitals. Clinical knowledge and temporal knowledge about antibiotic administration contained in the QUAIC antibiotic treatment guidelines are modelled in the prototype ontology. A rule-based reasoning system, which is used to answer clinical questions with regard to antibiotic administration in the guidelines, is also developed in the prototype ontology. The temporal reasoning part of the system is used to find administered antibiotics, dose intervals, dose durations and exact temporal relations between antibiotic administrations, and check if inconsistent temporal relations exist in the guideline ontology system. The international standard medical terminology SNOMED CT is also leveraged to represent the medical concepts in the guideline regimen recommendations to facilitate the medical terminology interoperability.

1.4 Thesis Organisation

Chapter 1 describes the background in the research area of OWL-based clinical guideline systems, the temporal knowledge representation issue encountered in this area and the research contribution to this issue. Chapter 2 describes and analyses the related research in the computerised clinical guideline system. This chapter describes some major approaches in this area such as Arden Syntax, Gliff, PROforma and Asbru etcetera. It also describes and analyses three main W3C ontology languages for knowledge representation and reasoning, the advantages of OWL ontology language over RDF and RDFS ontology languages, and the major applications of OWL in the medical domain. In addition, this chapter analyses the temporal knowledge representation issue in OWL and the OWL-based clinical guideline system, the current major temporal knowledge representation approaches in OWL and their advantages and disadvantages. Chapter 3 analyses various temporal constraints in clinical guidelines. In this chapter, an extended 4D fluent ontology is presented for modelling the temporal knowledge involving the valid calendar time, interval, duration, the repetitive or cyclical temporal constraints and the temporal relations such that it can enable the temporal knowledge related reasoning in clinical guidelines. Chapter 4 demonstrates the extended 4D fluent modelling method in an OWL-based antibiotic treatment guideline ontology which is derived from the "Intensive Care Unit Empirical Antimicrobial Treatment Guidelines" of QUAIC. This chapter provides a detailed analysis of the clinical knowledge and its temporal constraints contained in the antibiotic regimen recommendations provided by

the QUAIC antibiotic treatment guidelines. Classes, relations and attributes about diseases and drugs in the regimen recommendations are modelled in a domain ontology whereas classes, relations and attributes about temporal constraints in the regimen recommendations are modelled in an extended 4D fluent ontology. Chapter 5 presents a clinical knowledge reasoning system for the prototype ontology. It includes a non-temporal reasoning part and a temporal reasoning part. The non-temporal reasoning part contains the reasoning rules and functions for finding the antibiotic regimen recommendations provided by the QUAIC antibiotic treatment guidelines. The temporal reasoning part contains the reasoning rules and functions for finding administered antibiotics, computation of dose interval and dose duration, dose interval and duration compliance checking, and finding the temporal relations between administered antibiotics. The temporal relation reasoning is based on the Constraint Propagation Algorithm of Allen's interval algebra. Chapter 6 describes the evaluation procedure for the antibiotic treatment guideline ontology and analyse the evaluation results. The evaluation procedure consists of two parts. One part is the evaluation of the logical consistency of the ontology whereas another part is the evaluation of clinical question answering in the ontology. The evaluation of clinical question answering is based on a set of clinical questions which are often asked by health professionals with regard to the QUIAC antibiotic treatment guidelines. An evaluation matrix is also developed in terms of these questions. The dataset for the evaluation is based on both a synthetic patient dataset and a real patient dataset which is extracted from the open source Multiparameter Intelligent Monitoring in Intensive Care (MIMIC II) Database. The last chapter summarises the research contribution of the extended 4D fluent temporal knowledge modelling in OWL-based clinical guideline system, the limitations of this approach and possible future work.

Chapter 2 Literature Review

This chapter describes and analyses the related research in the computerised clinical guideline system. It includes the non-OWL based clinical guideline formalisms such as Arden Syntax, Gliff, PROforma and Asbru. It also describes and analyses three main W3C ontology languages for knowledge representation and reasoning, the advantages of OWL ontology language over RDF and RDFS ontology languages, and the major applications of OWL in the medical domain. In addition, this chapter analyses the temporal knowledge representation issue in OWL and the OWL-based clinical guideline system, the current major temporal knowledge representation approaches in OWL and their advantages and disadvantages.

2.1 Non-OWL Ontology Language Based Computerised Clinical Guideline System

Clinical guidelines and its more specific version, clinical protocols, are important clinical documents, which are the key tools for improving the quality of health care. However, the clinical guideline is traditionally a free text document where the clinical knowledge is stored in the unstructured format. As more and more medical knowledge is added to the free text clinical guidelines due to the increasing clinical findings, it often leads to a significant information overload for busy clinicians. As a result, it reduces the accessibility of guidelines for them [3]. Therefore, manually implementing clinical guidelines will prevent the distribution and implementation of guidelines in the daily clinical practice. Subsequently, it will reduce the efficiency of clinical decision making in daily health care.

Significant research from the fields of information technology and health care institutes has been devoted to the development of formal and machine manipulative representations of the medical knowledge in clinical guidelines. It is often called the computerised clinical guideline, the computer-based clinical guideline or computer interpretable guideline (CIG). As a kind of clinical decision support system, a computerised guideline system can assist clinicians to make efficient decisions, review and research their clinical practice in the daily care with regard to the guidelines.

There are already different models and formal languages developed to represent the clinical guidelines in a computer interpretable and manipulative format. According to De Clercq et al. in [4], many CIGs are designed in terms of the task network model (TNM) which models the guideline control flow as a network of specific tasks such as clinical decisions, plans or actions in a step-by-step manner. The formal language realises the underlying TNM of clinical guidelines in its vocabulary, syntax and semantics. Five major CIG formalisms analysed in [4] are Arden Syntax, GLIF (The Guideline Exchange Format), EON, PROforma (Proxy and Formalize), and Asbru respectively. Representation primitives or language constructs of each formal language are not the same but usually have some common ones such as plan, action and decision [5]. Each formal language consists of the control flow language and the expression language to represent knowledge types such as procedural knowledge and declarative knowledge contained in guidelines. The control-flow language usually specifies the structure (flow) of guideline tasks in terms of primitives of the TNM model, whereas the expression language usually describes the decision criteria which are in the body part of rules, i.e., the If part of rules [4].

However, unlike OWL which is a standard ontology language in the Semantic Web area, none of formalisms above has achieved a standard status in the CIG area. As described in [4], Arden Syntax uses the frame representation language to encode guideline knowledge in its knowledge slots which contain type, data, evoke, logic and action as the mandatory slots, and priority and urgency as the optional slots. The logic slot in Arden Syntax is used to specify the clinical decision criteria in production rules. GLIF previously used GEL (Guideline Express Language) which is based on Arden Syntax; but, it now uses an object oriented expression language GELLO to specify its decision criteria. The control flow languages of GLIF and EON are based on the RDF (Resource Description Format), which is the least expressive ontology language and lacks reasoning support in the ontology language family. The TNM of PROforma is defined in a task ontology which is rather a conceptual model and has four reusable task classes, i.e., plan, decision, action and enquiry. The values of attributes of each task can be entered into slots during the guideline knowledge acquisition stage. However, the formal language for its task ontology model is not based on any of the formal ontology languages such as RDF, RDFS or OWL. In PROforma, the formal language is a time-oriented control flow language- R^2L (Red Representation Language) to represent its control flow

structure of guideline tasks and the decision criteria. During the execution time, the language R^2L is translated into another language-L_{R2L} (Logic of R^2L) which is based on the predicate logic. The control-flow language and expression language for Asbru TNM are defined in XML.

As is stated in [6], the translation from the text-based guidelines to the machine interpretable and executable computerised guidelines in terms of the CIG formalisms above is cost expensive. The proprietary guideline execution engine of each CIG approach also prevents its wider application in the development of practical computerised clinical guideline system. In general, each CIG approach analysed previously has not moved beyond its development environment.

2.2 OWL-Based Computerised Clinical Guideline System

In contrast to the knowledge representation languages used in the CIGs previously analysed, there is also a trend that leverages the rich expressiveness and powerful reasoning capability offered by ontology languages to model and formalise medical knowledge contained in clinical guidelines.

2.2.1 Three Major Ontology Languages for Knowledge Representation and Reasoning

The term “ontology” originates from philosophy and denotes such a field which studies the metaphysics about the “nature of existence and categorical structure of reality” (The Oxford Companion to Philosophy 2005, page 670) [7]. A categorical scheme typically organises entities or things from the top level to the lower level in a hierarchical structure. Applying this philosophical ontology concept into computer science, it often refers to the controlled vocabularies which talk about concepts and relations and are used to classify things in the domain of interest. However, in the Semantic Web area, ontology is defined more strictly as a language referring to an explicit and formal specification of shared conceptualization of a domain of discourse [8].

In the Semantic Web research field, the World Wide Web Consortium (W3C) proposed three major ontology languages including Resource Description Framework (RDF), Resource Description Framework Schema (RDFS) and Web Ontology Language (OWL) for representing knowledge existing in the World Wide Web. Ontology provides a formalised vocabulary to precisely describe domain concepts and the relationship between these concepts in a machine accessible and manipulable format such that some intelligent applications can leverage these representations to draw useful knowledge from the web. Although these ontology languages are initially developed for representing knowledge on the Web, they are not limited to the contents on the Web and have been used widely to model knowledge in various domains.

RDF

According to W3C [9] [10], RDF is a knowledge modelling language used to semantically describe resources on the Web by using metadata such as the title, author and date of a web page. It provides a simple data model for representing web resources and their

relations in binary properties. The core language constructs of RDF are resource, property and statement. A resource can be an object of any kind on the Web which is identified by a URI (Uniform Resource Identifier) or an IRI (Internationalized Resource Identifier). The latter is a generalisation of URI and is used in the latest RDF recommendation. A property is a binary predicate to link resources in a triple which has a subject-predicate-object format and is called a statement. A statement can be visualised in a directed RDF graph in which the labelled nodes represent the subject and object and the arc represents the binary property or relation.

RDFS

Although RDF properties are for representing relations between resources, RDF does not provide vocabularies for describing these properties and the relations between these properties and other resources. As a semantic extension of RDF, the RDF vocabulary description language RDFS provides mechanisms for describing groups of related resources and the relationships between these resources [11]. The basic modelling primitives of RDFS include class, subClassOf, subPropertyOf, domain and range restrictions etcetera. RDFS together with RDF provide a mechanism to organise and interlink data in a relatively simple hierarchical and categorical ontology structure.

OWL

RDF and RDFS provide the means to represent knowledge in structured ontologies, but the expressivity of RDF and RDFS languages and related reasoning support are very limited. Reasoning capacities in RDF and RDFS are basically restricted to the inference in type, subClassOf, subPropertyOf, domain and range. As analysed in [12], some important reasoning features missing in RDF and RDFS include reasoning in local scope of properties, disjoint relation between classes, equivalent relation between classes, combination of classes such as union, intersection and complement, cardinality restrictions on properties, and property characteristics such as transitive, symmetric, reflexive, inverse and functional.

With regard to the demand for the richer ontology languages for knowledge representation, W3C developed the OWL ontology language family which includes three sublanguages: OWL Full, OWL DL and OWL Lite. However, OWL DL already gained the wider support than the

other two sublanguages since it has a better balance between the language expressivity and the reasoning capability, i.e., it is not only expressive but also decidable. The recent development of OWL is OWL 2 profiles which contain OWL 2 EL, OWL 2 QL and OWL 2 RL. These three sublanguages are the syntactic subsets of OWL 2 DL and vary in terms of the expressivity and reasoning capability.

OWL provides a set of much richer language constructs than the ones in RDF and RDFS to describe concepts in the domain and the relations between these concepts. The core OWL language constructs include class, individual, object property, data property, class expression construction in terms of union, intersection or complement of other classes and enumeration of individuals, property restriction, cardinality restriction on property, property chain, axioms for specifying relations between class expressions in terms of subclass, equivalent, disjoint and disjoint union relations, axioms for characterising and specifying relations between object property expressions in terms of sub-object property, equivalent, disjoint, inverse, domain and range of object property, functional, inverse functional, reflexive, irreflexive, symmetric, asymmetric and transitive characteristics of properties, and axioms for data property expressions in term of sub-data property, equivalent, disjoint, functional characteristics, and domain and range of data property [13].

OWL is based on Description Logic (DL) which has a set of constructors and axioms for representing knowledge in various domains and a well-defined formal semantics to precisely specify the meaning of each constructor and axiom. As stated in [14], large parts of OWL DL can be considered as a syntactic variant of the fragment of DL-SROIQ. For instance, axioms in SROIQ are divided into ABox (assertional axioms), TBox (terminological axioms) and RBox (relational axioms). The ABox axioms include concept assertion *C(a)*, role assertion *R(a, b)*, individual equality $a = b$ and individual inequality $a \neq b$. The TBox axioms include concept inclusion $C \sqsubseteq D$ and concept equivalence $C \equiv D$. The RBox axioms include role inclusion $R \sqsubseteq S$, role equivalence $R \equiv S$, complex role inclusion $R1 \circ R2 \sqsubseteq S$ and role disjointness *Disjoint (R, S)*. Symbols *C* and *D* in these axioms denote the concepts, whereas *a* and *b* denote the individuals; *R* and *S* denote the roles or relations. All of axioms in SROIQ can be precisely interpreted using Model-theoretic semantics. In Model-theoretic semantics, an interpretation I consists of a domain of I (Δ^I) and an interpretation function $\cdot^I$

^{I} which assigns each atomic concept A to a set $A^I \sqsubseteq \Delta^I$ and each atomic role R to a binary relation $\Delta^I \times \Delta^I$. Under this semantics, the concept assertion *C(a)* can be interpreted as $a^I \in C^I$; the role assertion *R(a, b)* can be interpreted as $< a^I, b^I > \in R^I$ and the role inclusion $R \sqsubseteq S$ can be interpreted as $R^I \sqsubseteq S^I$.

Based on the axioms and other DL constructors in SROIQ, OWL axioms about the relations between classes and individuals can be translated into the corresponding DL axioms. For example, the class assertion about what type an individual belongs to corresponds to the concept assertion in the ABox of SROIQ. Similarly, the assertions about relation, equality and inequality between two individuals correspond to the role assertion, individual equality and individual inequality in the ABox of SROIQ. The assertions about subclass of and equivalent relation between two classes correspond to the concept inclusion and concept equivalence in the TBox of SROIQ. The assertions about sub-property of and equivalent relation, property chain and disjoint relation between properties correspond to role inclusion, complex role inclusion and role disjointness respectively in the RBox of SROIQ. Therefore, the following property chain example expressed in OWL functional syntax SubObjectPropertyOf(ObjectPropertyChain(*:hasMother :hasSister*) *:hasAunt*) in [13] can be written as hasMother ∘ hasSister $\sqsubseteq$ hasAunt in DL.

In summary, the formal semantics of DL allows precise specification of the meaning of DL-based ontologies such that computer systems can exchange the ontologies unambiguously and can also make logical deduction to infer implicit knowledge from the explicitly stated facts in that ontology.

2.2.2 Major Applications of OWL in the Medical Domain

OWL has been widely used to model knowledge in different domains. Especially in the medical domain, many OWL-based medical ontologies have been developed, but most of them focus on the modelling of medical terminologies in the different areas of medical science. As mentioned before, SNOMED CT is OWL-based and one of the most comprehensive and largest clinical terminologies in the world. Moreover, many other OWL-based medical ontologies can be found in the BioPortal website of The National Centre for

Biomedical Ontology (NCBO). For example, Health Level 7 (HL7) Reference Information Model (RIM) focuses on the development of international medical information interoperability standards. Although HL7 RIM is initially developed in an object-oriented UML model, an OWL-based ontology version of HL7 RIM is also developed. GALLEN is an OWL-based comprehensive ontology which classifies several thousands of clinical concepts obtained from different medical domains and is open source and reusable. Gene ontology represents genes and gene product attributes in a species-independent manner and covers the areas of the associated biological processes, cellular components and molecular functions. Foundational Model of Anatomy (FMA) is an open source ontology concerned with the representation of human body structure.

2.2.3 The Application of OWL in the Computerised Clinical Guideline

There is also a trend which leverages the rich expressiveness and powerful reasoning capability of OWL to represent the knowledge contained in clinical guidelines such that it can produce a computerised clinical guideline to assist clinicians to make decisions. Compared to the non-OWL ontology based formalisms, OWL is a W3C standard modelling language which is supported by many tools such as Protégé, SWOOP, NeOn Toolkit and TopBraid Composer for ontology authorisation, visualisation and reasoning. Thus, it makes OWL as a competitive candidate for the computerised clinical guideline system.

In the development of OWL-based computerised clinical guidelines, some researchers focus on the development of a common ontology model, core vocabularies, architecture or methodology, whereas other researchers focus on the development of practical systems.

W3C Semantic Web for Healthcare and Life Sciences Interest Group (HCLSIG) has proposed a draft OWL ontology model for clinical guidelines which is called Adaptable Clinical Pathway and Protocol (ACPP) model and is similar to the TNMs in the non-ontology based formalism. This model leverages the declarative feature of OWL to adopt a prescriptive approach [15] which is different than the procedural approach often implemented in the non-OWL ontology based guideline systems. HCLSIG attempts to define a core set of vocabularies in that model which are the most common concepts in clinical guidelines to organise various clinical tasks and processes, patient clinical states, and situation constraints such as context,

goals, and inclusion/exclusion criteria. Clinical tasks or processes can be activated when the necessary antecedent conditions are met. The ACPP model has been used by HCLSIG to model clinical guidelines in stroke management, coronary artery bypass graft, the management of patients with ST-Elevated myocardial infarction and Diabetes. However, like the previously analysed guideline formalisms, the ACPP model is still in its development stage.

Kashyap et al. in [16] proposes a general architecture for creation and maintenance of computerised guidelines. The architecture consists of a data repository, a rule engine, an ontology engine and a web server. The data repository stores patients' data in an electronic health record (EHR) which resides in a database management server. The data repository connects it with the rule engine and the ontology engine via adapters at runtime to answer queries. The rule engine executes declarative production rules for clinical decisions, and also manages changes and detects inconsistency in the rule base. The ontology engine uses an OWL-based classification engine for classification and subsumption inferences and inconsistency checking on the ontology classes. The web server is used to present the application contents and results to users. Inside this architecture, the clinical guideline model is similar to GLIF3 but it is written in OWL language. The guideline model decomposes the guideline into decisions, actions, patient state transitions and definitions. Definitions of clinical concepts are represented either in OWL axioms or in if-then rules and are managed in the ontology engine.

De Clercq [17] proposes a methodology for ontological representation of clinical guidelines. The methodology separates domain-specific knowledge and problem-solving method (PSM) in modelling guidelines. The primitive-based guideline representation formalisms such as Arden Syntax, PROforma and GLIF often use explicit primitives to construct the eligibility criteria, actions and decisions in clinical guidelines. As a result, domain knowledge is always intertwined with procedural knowledge. In contrast, the PSM-based approach separates the domain ontology and the PSM (the method ontology) so that it can facilitate the reusability and sharing of developed guidelines. Two PSMs are proposed in this paper, which are the relatively simple primitive PSM and the complex PSM respectively. A method library contains all methods which represent primitive PSMs and complex PSM to solve the tasks

required by the guidelines. The method manager maps concepts from the domain ontology onto the knowledge roles in the method ontology to output the results triggered by the rulebase in the runtime such that it can provide recommendations for clinicians. However, the implementation of this approach in several actual cases shows that guidelines which are more complex or more domain specific are not very suitable for the PSM-based approach which is often too general for these guidelines.

In contrast to the previous efforts which define a general model, architecture or a methodology for guideline ontology, other researchers focus on the development of the practical clinical guideline based systems in different medical domains using OWL. Casteleiro et al. in [18] present a service-based application for diagnosis and clinical management of diabetic retinopathy for health professionals who are not familiar with Semantic Web technologies. This application is based on OWL, OWL-S (OWL for web service), and Semantic Web Rule Language (SWRL) and is created in Protégé and its plug-in OWL-S editor. The application uses the modular ontology design methodology to create four ontologies which include the SWRC ontology, the Organization Extension ontology, the Document Extension ontology and the Data Set ontology. The SWRC ontology reuses some vocabularies in Dublin Core ontology such as title, date and creator to model the relationships between general key entities such as organisation and document. The Organization Extension ontology reuses some medical concepts in Unified Medical Language System (UMLS) to model the health care related organisations which extend the general organisation entity in the SWRC ontology. The Document Extension ontology extends the general document entity in the SWRC ontology and models the medical concepts contained in clinical guideline documents using medical concepts in UMLS. The Data Set ontology contains the patient data and SWRL rules encoded in OWL's XML Presentation Syntax for the input and output of the web services. In a user-friendly interface, this application provides end users with three major services: a patient identification service, a GL clinical information service, and a GL recommendation service.

Chen et al. in [19] presents a clinical guideline-based anti-diabetic drug ontology system which is developed in OWL and SWRL. The system aims to recommend suitable drugs and monitor contraindication and side effects for general practitioners through a set of user

defined SWRL rules executed in a JESS rule engine in the Protégé environment. Abidi in [20] developed a guideline-based breast cancer follow-up care ontology system to provide patient specific recommendations for breast cancer patients. The follow-up guideline ontology is based on the Guideline Representation Model (GEM) and is developed in Protégé, but the rules for finding recommendations are written in the CPG Rule Syntax of GEM and executed in GEM execution engine. Romero et al. in [21] developed an ontology-based expert system which can automatically take patient information such as vital signs and current drug infusion rates from the patient monitor as input and produce the treatment recommendations for patients in the cardiac intensive care units (CICU). The ontology construction is guided by the knowledge based system development methodology CommonKADS (Knowledge Acquisition and Documentation Structuring) and is developed in Protégé and SWRL.

Different from the guideline systems which focus on the single diseases, Abidi et al. in [22] presents a COMET (Co-morbidity Ontological Modelling & Execution) guideline system which can support patients with comorbidities such as comorbid chronic heart failure and atrial fibrillation. Therefore, the authors in this paper focus on the merging of multiple clinical guidelines and pathways in the OWL-based guideline ontology. The major challenge encountered in their work is the reconciliation and alignment of the interventions recommended by individual guidelines and pathways without losing clinical appropriateness, patient safety and task pragmatics in the ontology. This involves the conceptual mapping between individual guidelines and pathways in order to integrate them in one comorbid pathway ontology. The ontology is developed in Protégé and is verified and validated through the Pellet reasoner and external medical experts for ensuring the concept consistency, satisfiability, conciseness, and correctness.

2.3 Temporal Knowledge Representation Issue and Modelling Approaches in OWL and the OWL-Based Clinical Guideline System

Although OWL has been successful in many knowledge-based applications, there are still some important knowledge types which cannot be directly modelled in OWL. Knowledge containing uncertain, propositional attitudes, epistemic and deontic modalities often involves predicates with arity more than two which are beyond the scope of binary predicate-based OWL. Knowledge with temporal constraints is one of such knowledge type which has a ternary predicate logic form which cannot be represented directly in OWL. Temporal knowledge is largely ignored in many OWL-based guideline systems analysed previously. The lack of support of temporal knowledge representation brings a major challenge to the wider adoption of OWL in the knowledge-based system including the computerised clinical guidelines.

The general logic form of temporal knowledge is the ternary predicate *R (a, b, t)* where the relation R between the individuals a and b holds at the temporal entity t. Ternary predicates cannot be represented in OWL. The meta-logic form *holds (R (a, b), t)* is also not supported by OWL since reasoning about relations over relations is undecidable in OWL [23]. However, temporal knowledge is an essential and indispensable part of various knowledge domains. Ignoring the representation of temporal knowledge in OWL will hinder the wider adoption of OWL in the knowledge based systems. In order to deal with this issue, researchers have proposed different solutions. According to O'Connor and Das in [24], these solutions can be divided into two types of approaches: the DL-based approach and the user-level based approach.

2.3.1 Temporal knowledge modelling approaches in OWL

2.3.1.1 DL-Based Approach

The DL-based approach is a fundamental solution of the temporal knowledge representation issue in OWL. It attempts to modify the underpinning Description Logic of OWL to develop a temporal description logic system such that a temporal OWL ontology language can be developed from it. This approach is basically based on the combination of a

subset of Description Logic with a kind of temporal logic. According to Artale and Franconi in a survey in [25], various combinations differ from each other in the aspects which include the adoption of an interval-based or a point-based notion of time, the way of handling of explicit or implicit temporal information, and external or internal view of explicit temporal information. The external view of time separates an individual into a static atemporal part and a temporal part. The temporal part of an individual describes the various states of the individual as "snapshots", i.e., the dynamic aspects of the individual at different times. In contrast, the internal view of time treats an individual as the collection of its distinct temporal parts which actually are the indispensable and internal components of the individual and hold at different moments.

Four types of temporal description logic are discussed in this paper. The interval-based temporal description logic usually follows the external approach to extend one of static description logics with an interval-based explicit time, whereas the point-based temporal description logic often follows the external approach to extend a kind of description logic with a point-based explicit time. For the interval-based temporal description logic, the full fledged interval-based logic is undecidable. For example, Schmiedel's formalism is very expressive, but it is undecidable and lacks computational machinery. The interval-based description logic proposed by Halpern and Shoham is also undecidable. Some fragments of the interval-based description logic (e.g., TL-ALCF proposed by Artale and Franconi) have been proved decidable and are interesting for applications. However, the expressivity of these fragments is seriously restricted. For the point-based temporal description logic, $CIQ_{U,S}$ is the most expressive and decidable one when having temporal operators on concepts and formulae. However, it will become undecidable when having temporal operators on the role side. The third type takes the internal view of time to add a temporal part, i.e., a temporal concrete domain to description logic. The most important work in this area is ALC(D) proposed by Baader and Hanschke. ALC(D) adds an admissible concrete domain D (i.e., the set of rational numbers with the comparison operators $<$, $\leq$, $=$, $\neq$, $\geq$, and $>$) to the description logic ALC while it still maintains the decidability. Time intervals and Allen's basic temporal relations can be converted to the operation in this concrete domain. Based on ALC(D), Milea et al. in [26] propose a temporal ontology language tOWL which extends the current OWL language to deal with temporal information. However, in order to

implement temporal reasoning in Allen's 13 basic relations, tOWL also adds a temporal reference layer and a 4D fluent layer on top of the concrete domain layer to represent temporal entities. The latter two layers in tOWL are realised by employing a temporal knowledge modelling method- 4 dimensional fluent (4D fluent) which is analysed in the next section "User-Level Based Approach". The tOWL ontology language has been tested in the financial application-Leveraged Buyouts (LBO) and the result appears to be promising. In contrast to the first three approaches, the fourth type only limits itself to deal implicit temporal information such as ordering, repeating and looping in a state-change based description logic to model plan-like knowledge. Therefore, the application of this approach is very restricted. Examples in this area are CLASP system (CLAssification of Scenarios and Plans) and RAT system (Representation of Actions Using Terminological Logics). In general, the DL-based approach is theoretically attractive, but a critical issue is how to develop a temporal description logic which is reasonably expressive and also decidable in reasoning. Unfortunately, such a practical temporal DL system has not emerged yet. Consequently, there is no recommendation for the related OWL language from W3C and practical tools for temporal knowledge representation and reasoning.

2.3.1.2 User-Level Based Approach

In contrast to the DL-based approach, the user-level based approach is more practical and relatively easy to implement in OWL. The user-level based approach does not modify the underpinning logic of OWL but represents the temporal knowledge within the existing OWL language constructs by leveraging a representation method. Three major representation methods of this approach proposed are RDF reification, N-ary relation reification, and the 4D fluent temporal knowledge modelling methods.

RDF Reification

The RDF reification is a general mechanism of making statements about statements, i.e., describing other RDF statements using RDF to record the information about the statements such as when statements were made, who made the statements and other similar information [9]. Each reified statement is an instance of the type rdf:Statement and has a subject and an object denoting the participating entities in that relation, a predicate denoting the relation and other extra information such as the temporal information about

the statement. In order to describe the reified statement in other statements, a URI (e.g., http://www.example.com/) is assigned to it as ex:statement123 where “ex” is the prefix of the URI. For example, Mary was administered with vancomycin at 9:00 am on15 October 2010. This statement could be reified as the following set of RDF statements.

ex:statement123 rdf:type rdf:Statement .

ex:statement123 rdf:subject ex:mary .

ex:statement123 rdf:predicate ex:administeredWith .

ex:statement123 rdf:object ex:vancomycin .

ex:statement123 ex:time “2010-10-15T09:00:00”^^xsd:dateTime .

The reified statement is shown in the following RDF graph in Figure 1.

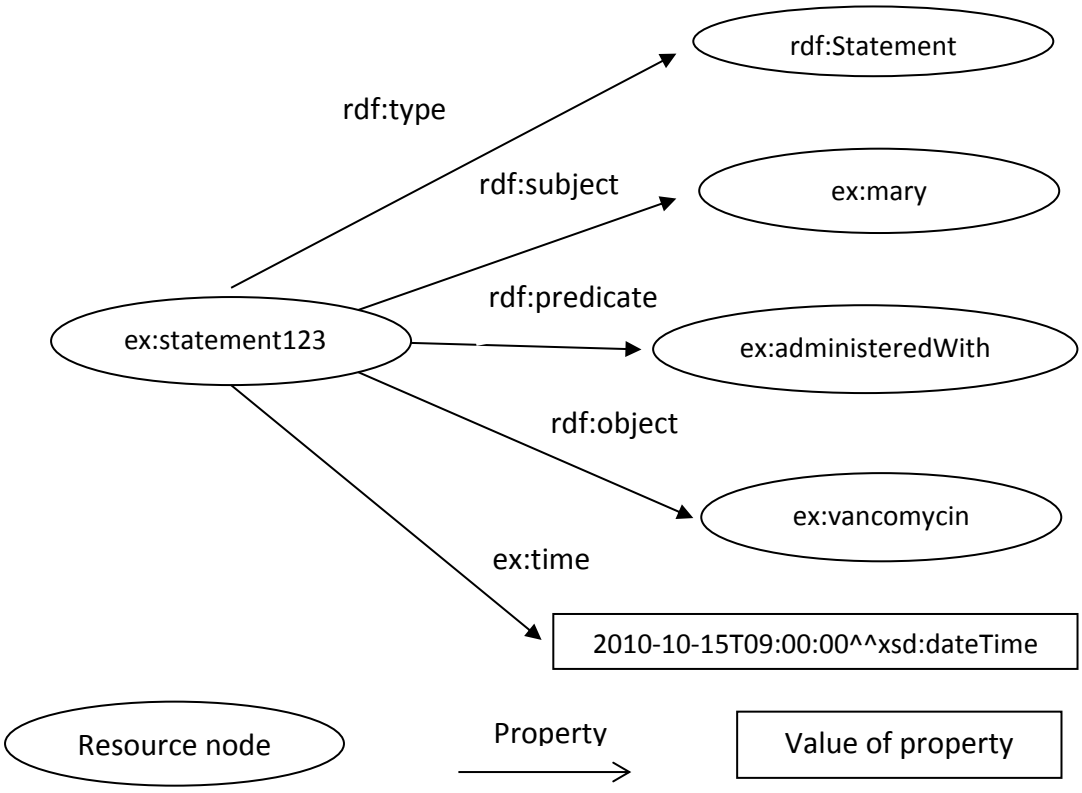


Figure 1. An Example of RDF Reification

However, RDF reification only deals with statements and their subjects, objects and predicates rather than the real relations between entities. Thus, it is not semantically natural due to the treatment of relations as statements and has no OWL reasoning support in terms of the property characteristics of relations such as transitive, symmetric, inverse and functional. For example, the relation “administered to” as the inverse of “administered with” relation describes to whom a drug is administered. The inverse relation between

these two relations does not hold any more due to the original relation “administered with” is reified as an RDF statement. Therefore, there is no reasoning support for the inverse relation. Moreover, RDF reification has the disadvantage of object proliferation since more statements are added in the ontology.

N-ary Relation Reification

According to W3C in [27], the N-ary relation reification is general method to represent predicates with higher arity in ontologies. It converts the relation to a new class in the ontology. Each instance of that class itself has binary relations connecting the participating entities in the original relation. Temporal information is therefore bound to the instances of the new class. As to the previous example, the original relation “administered with” could be converted into a new class namely “DrugAdministration” and then reified as the followings triples:

ex: DrugAdministration rdfs:subClassOf ex:ReifiedRelation
ex:drugAdministration1 rdf:type ex:DrugAdministration .
ex:mary rdf:type ex:Patient .
ex:vancomycin rdf:type ex:Drug .
ex:drugAdministration1 ex:has_patient ex:mary .
ex:drugAdministration1 ex:has_drug ex:vancomycin .
ex:drugAdministration1 ex:admin_time “2010-10-15T09:00:00”^^xsd:dateTime .

From the above triples, an ontology could be obtained as shown in Figure 2.

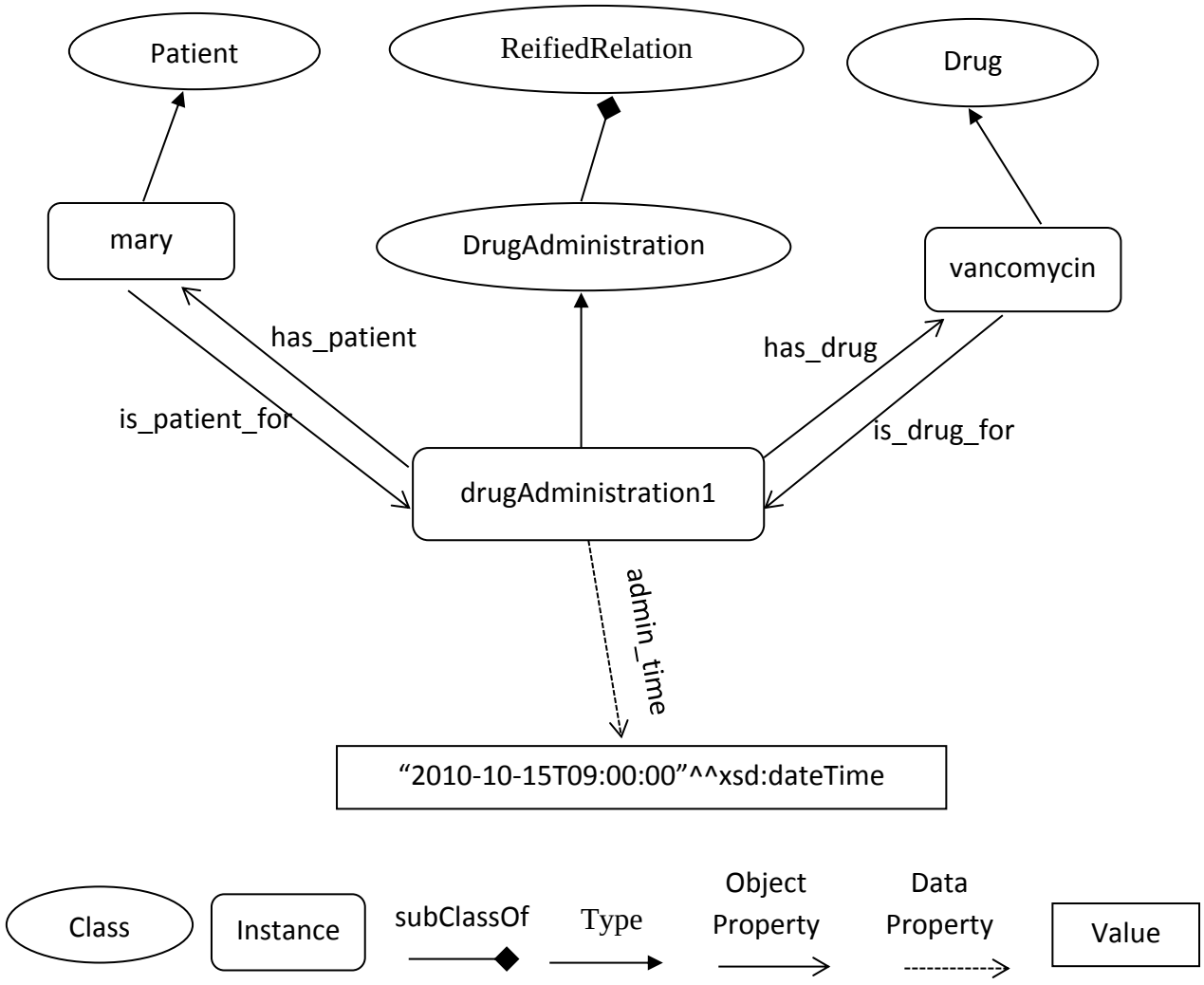


Figure 2. An Example of N-ary Reification

N-ary reification approach is semantically more natural than RDF reification since it deals with relations rather than statements. O'Connor and Das in [24] developed an N-ary reification based valid-time temporal model that can be reused in different OWL-based applications. This model has a root class called temporal:Fact for modelling all entities which are the reified binary relations holding in time. Instances of class temporal:Fact connects temporal entities such as time instant or interval via the property temporal:hasValidTime. A library of methods containing a set of user defined temporal predicates are also developed using SWRL built-in mechanism to implement 13 Allen's interval-based temporal operators. There are more than 20 built-ins for date, time and duration in the core set of the library for writing temporal rules. An associated temporal query language namely SQWRL (Semantic Query-Enhanced Web Rule Language) is also developed and implemented in Protégé.

Shankar et al. in [28] have applied the N-ary reification based temporal model in their ontological framework namely EPOCH proposed in [24] which is used for clinical trial management with regard to a clinical trial protocol.

Tao et al. in [29] propose a temporal ontology named CNTRO (Clinical Narrative Temporal Relation Ontology) for modelling unstructured temporal knowledge in clinical narratives. In contrast to the ontological modelling for structured, valid and absolute temporal data in databases, CNTRO is mainly for modelling unstructured temporal data in clinical texts. Therefore, it defines two special classes (i.e., "TimePeriod" and TimePhase") for modelling periodical time interval in clinical notes. It is also allowed to model relative time, uncertainty and temporal relations between clinical events without specifying the time stamp of these events. Like SWRL temporal ontology proposed by O'Connor and Das in [24], CTNRO is also an N-ary relation reification-based temporal ontology. Moreover, it defines a "TemporalRelationStatement" class which is based on the RDF-Reification modelling method analysed previously to represent temporal relation between two events by defining the subject, predicate and object.

However, as analysed in [23], [27] and [30], the N-ary reification approach prevents the use of many OWL operators for reasoning such as inverse, symmetric, transitive, functional and inverse functional. For example, it suffers data redundancy in terms of reasoning over the inverse of relations. As can be seen in Figure 2, in order to reason over the relation "administered to" (the inverse of "administered with" relation) to find to whom the drug vancomycin was administered, two extra inverse properties (i.e., "is_patient_for" and "is_drug_for") have to be added into the ontology. Therefore, the reasoning process has to take account of these inverse relations to find the patient to whom the drug was administered. It is also very awkward to specify the local range and cardinality restrictions on properties since the original relation is reified as a new class and the related semantics of the original relation is not applicable anymore. The domain (i.e., "Patient") and the range (i.e., "Drug") of the original relation "administered_with" do not hold anymore due to the reification. In addition, it suffers object proliferation like RDF reification since a new class and instances of this class are created due the reified relation. For example, a new instance of "DrugAdministration" will have to be created if the patient or drug changes.

4D Fluent

Welty and Fikes in [23] propose a temporal knowledge modelling method for OWL that is called 4D fluent. This approach is closely related to the four-dimensionalism philosophy. The traditional three dimensionalism philosophy views entities in the wold as three dimensional (i.e., length, width and height) and temporally non-extended objects. It means that objects last with different properties over different times but are still identified as the same objects through the whole period at which they exist. That is to say the object endures by being wholly present at each moment at which they exist [31]. In contrast to this three dimensional view, the four dimensionalism views entities as the aggregates of their distinct temporal parts and none of them are identical with the whole "space-time worm" concatenating these temporal parts [31]. According to this view, all entities from the whole universe to a single physical object are the four-dimensionally extended wholes which last over time without being wholly present at every time at which they exist but have distinct temporal parts (i.e., time slices of the space-time worms) at each moment. The 4D view is similar to the internal view of time in [25]. According to the internal view, the different states of an individual are seen as different individual components. As a result, an individual is a collection of distinct temporal parts and each of these temporal parts holds at a particular moment.

The 4D fluent method applies the four dimensionalism philosophy to model temporal knowledge in OWL. The concept "fluent" denotes the binary relation that holds within a certain time interval and not in others [23]. In 4D fluent, a relation between two entities which holds in a time instant or interval can therefore be represented as the relation of their temporal parts which are bound to the same temporal entity. Similarly, the attribute of an entity becomes the attribute of the temporal part of the entity. Consequently, the 4D fluent representation method yields a reusable high level 4D fluent ontology [23] which contains the following classes and properties.

- Time slice class or temporal part class (e.g., TimeSlice) which holds the temporal parts of all participating entities in binary fluents.

- Temporal entity class (e.g., TemporalEntity) which includes a time interval subclass (e.g., TimeInterval) to hold the individual interval entities and a time instant subclass (e.g., TimeInstant) to hold the individual instant entities.
- An object property such as “hasTemporalEntity” connecting the temporal part with its temporal entity.
- An object property such as “hasTemporalPart” connecting the participating entity with its temporal parts.

As to the previous example, the relation “administered with” between Mary and the drug Vancomycin can be converted to the relation between the temporal parts or time slices of Mary and Vancomycin in terms of the 4D fluent ontology. Thus, the following triples could be obtained based on the 4D fluent ontology.

ex:TimeSlice rdf:type owl:Class .
ex:mary rdf:type ex:Patient .
ex:vancomycin rdf:type ex:Drug .
ex:mary@t1 rdf:type ex:TimeSlice .
ex:vancomycin@t1 rdf:type ex:TimeSlice .
ex:mary ex:hasTemporalPart ex:mary@t1 .
ex:vancomycin ex:hasTemporalPart ex:vancomycin@t1 .
ex:mary@t1 ex:administeredWith ex:vancomycin@t1 .
ex:mary@t1 ex:hasTemporalEntity ex:t1 .
ex:vancomycin@t1 ex:hasTemporalEntity ex:t1 .
ex:t1 ex:timeValue “2010-10-15T09:00:00”^^xsd:dateTime .

The visualised 4D fluent ontology structure is shown in Figure 3.

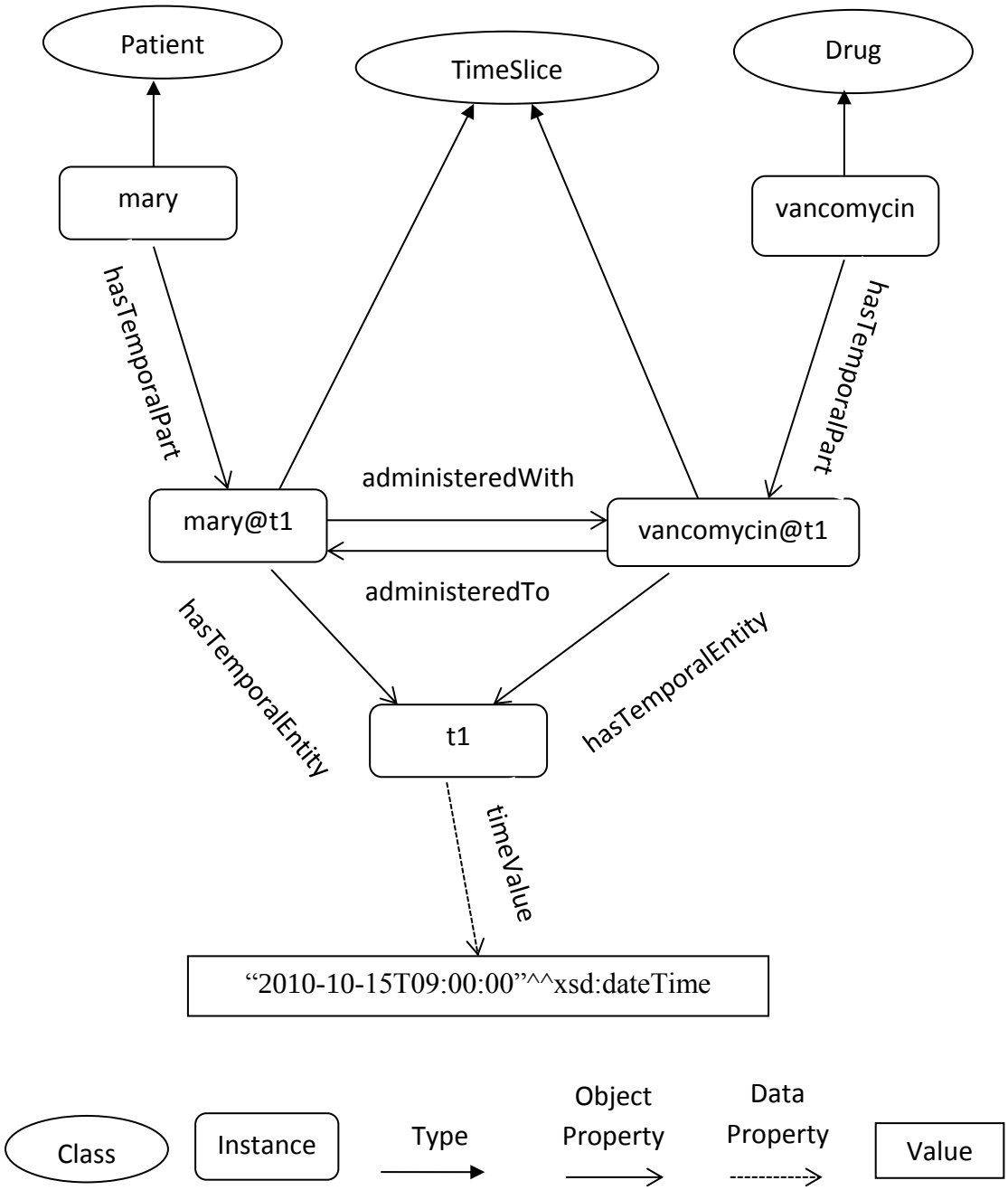


Figure 3. An Example of 4D Fluent

The 4D Fluents ontology in [23] imports concepts from the OWL-Time ontology which provides rich descriptions of temporal data such as intervals, instants, durations, and valid calendar time [32]. However, the representation of OWL-Time is not lightweight. It is neutral to temporal knowledge modelling methods and only focuses on the descriptions of temporal data [24]. Like RDF reification and N-ary relation reification, 4D fluent also suffers the proliferation of objects and requires the rewriting of the source ontologies. However, the major advantage of 4D fluent over other approaches is that it maintains full OWL

expressiveness and has better OWL reasoning support [30]. OWL has devoted many of its language constructs to express binary relations. Unlike reification, the original relations modelled in 4D fluent will not lose. Thus, the related semantics of the relations is still maintained in the ontology. The only concern is that the domains and ranges of the binary fluents need to be adjusted to the temporal part or time slice class. Moreover, it has less data redundancies in inverse, symmetric, transitive, functional and inverse functional property characteristics. For example, the inverse of "administered with", i.e., "administered to" is only added to the ontology once as shown in Figure 3.

The 4D fluent ontology has been used in different OWL-based applications to deal with events or activities with a temporal constraint that is a valid calendar time point (e.g., Gary bought a laptop on 03-08-2002) or an interval between two calendar time points (e.g., John worked for the company ABC from 01-10-1990 to 20-09-1996). In other researchers' work [30] [33], the 4D fluent ontology is enhanced with qualitative temporal interval (i.e., interval with the values of both start point and end point unknown) and semi-quantitative interval (i.e., interval with either the value of start point or the value of end point unknown). Temporal reasoning for finding temporal relations between events in these applications is usually realised in SWRL rules based on Allen's interval algebra.

For example, Okeyo et al. in [34] present a 4D fluent-based activity model-ADL (activities of daily living) in the smart home environment such as concurrent meal preparation. This model covers single activities, composite activities, static and dynamic aspects of activities, but particularly focuses on the composite (sequential, interleaved and concurrent) and dynamic activities since they involve the temporal constraints. Moreover, a set of inference rules which are based on 13 Allen's basic interval relations and are written in SWRL has been provided for composite activity recognition such as complex dependencies among activities. Krieger et al. in [35] present a temporal ontology in the MUSING project (Multi-industry, Semantic-based next generation business Intelligence). The temporal ontology has two top-level classes which are Perdurant and TimeSlice. The Perdurant class is used to encode all dynamic entities and the TimeSlice class is used to encode the temporal parts of these entities. Temporal relations connect time slices of these entities such that it can enable reasoning over temporal relations between these entities based on Allen's interval relations

and SWRL rules. Harbelot et al. in [36] propose a continuum model for objects which evolve over time in space in the Geographic Information System (GIS) domain. The continuum model represents the knowledge about the evolution of objects and their spatial-temporal relations. The temporal knowledge representation of objects is based on the 4D fluent modelling method. Inference on quantitative and qualitative temporal data is realised by the Allen's relations and SWRL rules. Moreover, Batsakis & Petrakis [30] developed a temporal knowledge query language-TOQL to handle both quantitative and qualitative temporal relationships in the 4D fluent based ontology. Evdoxios in [37] developed a tool to implement the queries using TOQL language.

2.3.2 The Temporal Knowledge Representation Drawback of OWL-Based Clinical Guideline System

Many non-OWL ontology based clinical guideline formalisms such as Arden Syntax, GLIF, PROforma, Asbru and CG_KRM support the temporal knowledge representation and reasoning to some extents. Arden Syntax supports the basic time instant based representation and offers a number of operators for extraction and reasoning of temporal information from clinical data. In Arden Syntax, the instant timestamp associated with patient records allows a range of simple temporal queries, whereas an interval timestamp associated with data needs more complex queries [24]. In GLIF, according to Terenziani et al. in [38], temporal constraints and relations are expressed by two types of temporal expression using the GEL language. The type of "times expression" specifies the number of times within an interval. The type of "every expression" specifies the fuzzy duration. Temporal reasoning based on temporal rules such as "occurs_at", '"is_before", "is_after", and "overlaps" can infer entailed temporal data. The guideline formalism CG_KRM (Clinical Guidelines Knowledge Representation Manager) proposed by Terenziani in [39] provides a set of constructs to represent various temporal knowledge in atomic clinical action and composite clinical action. Correspondingly, CG_EM (Clinical Guidelines Execution Module) provides a guideline engine to execute guideline knowledge represented in CG_KRM including temporal knowledge such as request time, reservation time, validity time, report time, transaction time, sequence relation, concurrency relation, alternative relation and cyclic actions. In PROforma, temporal constraints on the accomplishment of tasks, task duration and delays, and preconditions of actions can be defined in terms of each plan.

Temporal abstraction from raw clinical data in the reasoning is also supported. Asbru focuses on the representation of explicit declarative temporal aspects of intention-based durative skeletal clinical plans. As stated by Shahar et al. in [40], intentions can be viewed as temporal patterns related to health care provider actions or patient clinical states to be achieved, maintained or avoided. Clinical actions recommended by guidelines can be continuous. The execution order of clinical plans might be in parallel, sequence or a specified temporal order. Temporal scopes and parameters of guideline plans can be flexible. With regard to these issues above, Asbru uses time annotation to represent temporal knowledge contained in clinical guidelines such as uncertainty in starting time, ending time and duration of time intervals, multiple time lines, temporal shifts, and temporal repetitions. These major features make Asbru more time-oriented than other guideline formalisms.

Weng et al. in [41] propose a frame formalism-based temporal ontology for modelling patient scheduling tasks in clinical trial protocols which is implemented in protégé 2000. Since patient schedule is dynamically changing due to the changes of patient state, various temporal constraints need to be modelled in the frame-based ontology. In their temporal ontology, temporal constraints involving absolute calendar time, relativity, indeterminacy and cyclical pattern can be modelled and computed. A prototype scheduling decision-support tool for managing patient visit scheduling is developed. The tested results in dozen of clinical trial protocols show their ontology is able to produce patient-specific schedules with regard to these protocols.

In contrast to the non-OWL based guideline system, the representation of temporal knowledge is largely ignored in many OWL-based guideline systems due to the limitations of OWL. As analysed previously, The ACPP clinical guideline model proposed by HCLSIG does not contain concepts to model temporal knowledge in guidelines. The architecture for creation and maintenance of OWL-based guidelines in [16] and the methodology for creating OWL based guidelines in [17] do not propose a temporal knowledge representation method in their works. Similarly, the various OWL-based practical guideline systems described in [18], [19], [20], [21] and [22] also do not implement the temporal knowledge reasoning in their guideline ontologies. However, temporal knowledge is an indispensable

part of different knowledge domains. The lack of support of temporal knowledge in the OWL based guideline system will prevent its wider use in the daily health care practice.

In summary, the computerised clinical guideline systems including non-OWL based system and OWL based system and the main user-level based temporal knowledge representation methods are discussed in this chapter. Compared with the non-OWL ontology language based clinical guideline formalisms, OWL provides a standard ontology language for knowledge representation in clinical guidelines. However, due to the limitation of underpinning binary based predicate logic of OWL, temporal knowledge cannot be directly represented in OWL and has to leverage a representation method to model it in OWL. Among these methods, 4D fluent has better OWL reasoning support than other approaches. Considering these advantages, the 4D fluent method is focused for temporal knowledge representation of clinical guidelines in this research.

Chapter 3 Temporal Knowledge Analysis in Clinical Guideline and the Extended 4D Fluent Modelling Method

The previous analysis in Chapter 2 has shown that 4D fluent provides an effective representation method to model temporal knowledge. However, the original high level 4D fluent ontology are often limited to the relatively simple temporal constraints, i.e., the valid calendar time and interval. In many circumstances, temporal constraints in various domains especially in clinical guidelines tend to be more complex than the valid calendar time and interval. Therefore, extending the current 4D fluent ontology to model more complex temporal constraints contained in clinical guidelines is very necessary for the development of the practical OWL-based guideline systems. In this chapter, an extended 4D fluent temporal ontology is presented. The extended 4D fluent ontology can not only handle valid calendar time and interval, but also handle the more complex temporal constraint found in clinical guidelines.

3.1 Temporal Constraints Analysis in Clinical Guidelines

There are many types of temporal constraint which are more complex than the valid calendar time and interval. These temporal constraints often involve repetition, relativity, indeterminacy, delay, fuzziness and temporal relation and are often used together.

Events or activities are often repeated or cycled in a certain temporal pattern. It is not very difficult to find repetitive temporal events in daily life. For example, a university student attends a business lecture at 10am on every Tuesday in the first semester. This activity is repeated at a specific day and time. An activity can also be repeated at a periodic interval. For example, a business man travels to a city to buy products every 3 months. The periodic time or interval may have an extent of indeterminacy or uncertainty sometimes. For example, a country has a rainy season from around the mid of June to the mid of July each year. Moreover, the temporal constraints in repetitive events can also be relative. For example, a sportsman has an outdoor training schedule at 7am on day 1, day 2, and day 3 each week. This temporal constraint type is relative to the start time of each cycle.

The repetitive temporal constraint is also very common in the area of clinical guidelines. It often exists in clinical recommendations for drug dosage including dose interval, dose duration and dose frequency, drug experiment schedule of patients, medical examination of the patient's body or vital signs, and medical procedures such as blood culture collecting and blood glucose monitoring. The following examples were found in several clinical guidelines collected from local hospitals in New South Wales (NSW), Australia.

1. Less than daily subutex dosage: the frequency of dosing on Monday and Wednesday should be twice the individually titrated daily dose, and three times the individually titrated daily dose on Friday, with no medication on the intervening days.
2. Blood cultures for persistently febrile patients (e.g., neurosurgical) should be collected regularly (e.g., every 48 hrs) to detect line-associated sepsis.
3. For patient with sepsis and shock, add vancomycin 1.5g IV 12 hourly to provide MRSA cover.
4. For patient with suspected community-acquired meningitis and herpes simplex encephalitic picture, use acyclovir 10mg/kg IV 8 hourly for at least 14 days.
5. An asthma patient when discharged home, only requires bronchodilator every 3+ hours. Continue salbutamol (assess technique with MDI and Spacer). Consider prednisolone (usually 1mg/kg/day, max 50mg, for 3 - 5 days then cease).
6. For moderate asthma patients whose age are greater than 5 yrs, 12 puffs salbutamol via spacer every 20 minutes – up to 3 times, or up to 3 X 5mg every 20 minutes.
7. Frequency of blood glucose monitoring on diabetic type 1 patient: pre-breakfast (fasting), pre-midday meal, pre-evening meal, 2 hours post evening meal.
8. For all medical and surgical patients who are admitted to ICU/CCU and are not diagnosed with type I or type II diabetes, their BGL's are to be taken three times a day. The times are as follows: 0600, 1400 and 2200.

Among them, example 1 describes a medical event which should be repeated on specific days (Monday, Wednesday, and Friday). Examples 2, 3, 4 and 5 describe the medical events which should be repeated at a periodic interval; but, examples 4 and 5 have a duration constraint in the repetitive interval respectively. Moreover, examples 4 and 5 involve the temporal indeterminacy or uncertainty. Example 6 describes a medical event which should be repeated at a periodic interval with the frequency of dose specified. Example 7 and 8

describe the medical events which should be repeated in terms of schedule, but this temporal constraint type is relative and needs to be anchored to an absolute calendar start time of each cycle in order to produce a treatment schedule.

The constraint about the delay of an event is also an important type which can be found in various domains. For example, the meeting should start no more than 15 minutes after the end of morning tea. Moreover, events may involve fuzzy temporal constraints in some situations. For example, Mike was exercising regularly in a gym. However, if the values of fuzzy constraints cannot be determined, it is not possible to compute them to find the specific temporal pattern of the activities. In clinical guidelines, the temporal constraints about delay and fuzziness are also common. The following two examples found in clinical guidelines are about delay and fuzziness respectively.

9. Delay the first dose of buprenorphine until the patient shows significant features of withdrawal (usually more than 24 hours after the last dose of methadone).
10. Potassium levels should be monitored regularly and replaced promptly.

The analysis of these clinical examples is summarised in Table 1.

Clinical Example	**Temporal Constraint**	**Source**
E.g.1	Repetition at a specific day	S4
E.g.2	Repetition at a periodic interval	S1
E.g.3	Repetition at a periodic interval	S1
E.g.4	Repetition at a periodic interval with indeterminacy and duration	S1
E.g.5	Repetition at a periodic interval with indeterminacy and duration	S3
E.g.6	Repetition at a periodic interval with frequency	S3
E.g.7	Repetition in terms of relative schedule	S2
E.g.8	Repetition in terms of relative schedule	S2
E.g.9	Delay	S4
E.g.10	Fuzziness	S5

S1: Intensive Care Unit Empirical Antimicrobial Treatment Guidelines, QUAIC, NSW
S2: Blood Glucose Monitoring, Broken Hill Health Service, NSW
S3: Nurse Practitioner Clinical Practice Guidelines for the Management of Asthma, Sydney West Hospital, NSW
S4: Opioid Treatment Program: Clinical Guidelines for Methadone and Buprenorphine Treatment Space, NSW Government
S5: Guidelines for the Management of the Patient with Diabetes Ketoacidosis (DKA), RPA, NSW

Table 1. The Analysis of Temporal Constraints in Clinical Guidelines

As shown in Table 1, most constraints involve repetition in a certain temporal pattern. The repetitive or cyclical temporal constraint is of particular important for compliance checking in clinical practice with regard to guidelines such as drug administration where fixed periodical intervals between doses need to be followed for safety and efficacy purposes. Similarly, many clinical procedures such as monitoring of blood pressure, pulse, repository rate, temperature, blood glucose and creatinine clearance of patients in the intensive care unit (ICU) and emergency department also require following periodical intervals in practice for maintaining the procedure quality.

Temporal relation is also a very important and complex temporal constraint on events or activities in different domains. Temporal relation such as" before", "after" and "during" is intrinsic to various activities including clinical activities. Daily life activities always involve temporal order. For example, Ben went to a supermarket after work. In clinical guidelines, clinical plans are often arranged in terms of a certain sequential order or other temporal relations. For example, insulin therapy may be reduced or stopped until potassium has been replaced to prevent extreme hypokalemia if potassium levels of DKA patients are very low (S5). Temporal relations are sometimes implicitly stated in clinical guidelines. For example, the flowchart below displays a sequence of clinical plans in adult bowel management. Proper temporal arrangement of clinical activities in clinical guidelines is vital to the improvement of health care quality.

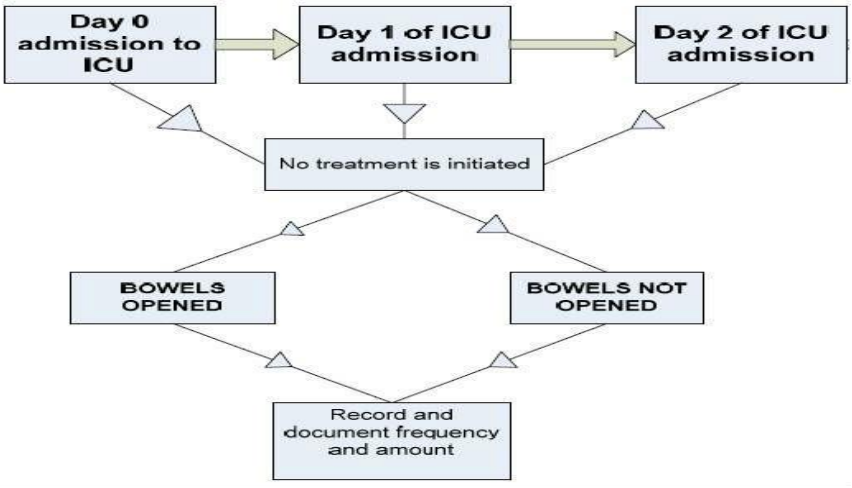


Figure 4. Intensive Care Adult Bowel Management Flowchart from Jon Hunter Hospital, NSW

3.2 The Extended 4D Fluent Ontology Analysis

In order to deal with the repetitive temporal constraint and the temporal relation constraint in the OWL-based clinical guideline system, the original 4D fluent ontology needs to be extended. In the original 4D fluent ontology, the major classes and properties include a time slice class, a temporal entity class which itself has two subclasses-time interval and time instant, an object property such as "hasTemporalEntity" for connecting the temporal part with its temporal entity and an object property such as "hasTemporalPart" for connecting the participating entity with its temporal parts. These classes and properties are reused in the extended 4D fluent ontology, but are extended in the following aspects (Figure 5):

- Rather than having two subclasses (i.e., time instant and time interval) under the temporal entity top class in the original 4D fluent ontology, the temporal entity top class is extended with five disjoint classes named "Time_Instant", "Time_Duration", "Time_Period", "Time_Interval" and "Repetitive_Temporal_Constraint" respectively as shown in Figure 5.
- The "Time_Instant" class is used to hold all individual valid calendar times in the original 4D fluent ontology. It is extended with three subclasses namely "Start_Time", "Following_Time" and "End_Time" respectively. The reason for extending the "Time_Instant" class is that clinical events or activities such as drug administration often have a start time, one or more following times and an end time. For example, a patient was administered a drug which started at time t1, followed by t2, t3, t4, and ended at t5. Clinical guidelines often require these activities to follow a fixed time interval and duration such as dose interval and duration for safety and efficacy purposes. Therefore, it is necessary to know the values of start time, following time and end time in order to compute the actual interval and duration of these activities.
- The "Time_Duration" class is used to record the length of time period of a clinical activity. The granularity of the time value depends on the knowledge domain and the requirements of applications. Therefore, seven more subclasses are created under this class which are "Duration_Years", "Duration_Months", "Duration_Weeks", "Duration_Days", "Duration_Hours", "Duration_Minutes" and "Duration_Seconds".

- The “Time_Period” class denotes a period that a clinical event or activity lasts from beginning to end. It has a start time and an end time and the length of the period of the event is the duration that the event lasts, i.e. the difference between the start time and the end time. Similarly to other researchers [30] [33], this class is extended with “Quantitative_Time_Period”, “Qualitative_Time_Period” and “Semi_Time_Period” three subclasses. The “Quantitative_Time_Period” is used to hold the temporal entities in which both the values of start time and end time are known. The “Qualitative_Time_Period” class is used to hold the temporal entities in which both the values of start time and end time are not known. The “Semi_Time_Period” class is used to hold the semi-quantitative temporal entities in which either the value of start time or the value of end time is known. Two subclasses are created under this class namely “Left_Close_Time_Period” (i.e., only the start time is known) and “Right_Close_Time_Period” (i.e., only the end time is known). Time periods related to the qualitative and semi-quantitative time period classes are very common in free text clinical records. For example, clinical notes such as patient progress notes often only chart the start time of a drug dose. Sometimes, the time information of a drug dose is not explicitly charted in the notes.
- “Time_Interval” class. The original 4D fluent ontology does not differentiate the concept “interval” from the concept “period”. However, in clinical guidelines, the concept “interval” often means the time period between two adjacent clinical events in a time sequence. For example, a dose interval between last dose time of an administered vancomycin and next dose time of the same drug for a patient. Therefore, it has two endpoints and the length of the interval is a difference between the two endpoints. Recommendations in clinical guidelines often have repetitive temporal constraints related to this interval type. Therefore, in the extended ontology, the “Time_Interval” class only denotes the period between the time points of two adjacent events.
- The “Repetitive_Temporal_Constraint” class is used to hold each periodical interval which is specified in a knowledge domain. There are two types of repetitive events according to Loganantharaj and Giambrone in [42]. One is the periodic repetitive event which repeats at regular intervals such as every 12 hours and every 2 days. Another one is the aperiodic repetitive event which repeats without regularity such

as random events. However, this type of repetition is very rare in clinical guidelines and is difficult for computation due to a lack of temporal patterns. Therefore, only periodical intervals are modelled in the extended 4D fluent ontology. Under the "Repetitive_Temporal_Constraint" class, it is the subclass "Periodical_Interval" which by itself has seven subclasses with different granularity to hold periodical interval instances of each type in clinical guidelines, which are "Every_X_Years", "Every_X_Months", "Every_X_Weeks", "Every_X_Days", "Every_X_Hours", "Every_X_Minutes" and "Every_X_Seconds" respectively. For example, an antibiotic regimen recommendation would have "every 12 hours" as the temporal constraint of vancomycin dosage for patients.

- The object properties defined in the original 4D fluent ontology are reused for the connection between the temporal part and its temporal entity and the connection between the participating entity and its temporal parts. In addition, two more object properties namely "open_instant" and "close_instant" are created for connecting a time period with its start time and end time or connecting a time interval with its open endpoint and close endpoint.
- Similar to other researchers' works in the modelling of temporal relations using the 4D fluent method [30][34][35][36], a set of object properties for representing Allen's interval relations (Appendix 5) are defined to deal with temporal relation reasoning in the ontology. However, the temporal relations defined in our ontology contain 27 Allen's relations which are based on the compositions of basic relations in the transitivity table of Allen's interval algebra. These temporal relations and the related temporal reasoning are analysed in Chapter 5.

In summary, the extended 4D fluent ontology enables the modelling of temporal knowledge involving valid calendar time, interval, duration, repetitive or cyclical temporal constraints and temporal relations. This then makes it possible to implement temporal knowledge related reasoning in the OWL-based clinical guideline system. In order to demonstrate how the extended 4D fluent ontology works in the OWL-based clinical guideline system, a prototype on antibiotic treatment guideline ontology system is built. The antibiotic treatment guideline ontology is analysed in the next chapter.

Figure 5. The Extended 4D Fluent Ontology Built in OWLViz Plugin in Protégé 4.1

Chapter 4 The Antibiotic Treatment Guideline Ontology Analysis and Design

The extended 4D fluent ontology is demonstrated in an OWL-based antibiotic treatment guideline ontology which is derived from the "Intensive Care Unit Empirical Antimicrobial Treatment Guidelines" written by QUAIC expert group in November 2010 for local NSW hospitals (http://intensivecare.hsnet.nsw.gov.au/state-wide-guidelines). This ontology is built with Protégé 4.1 and visualised in its plugins OWLViz and OntoGrap. The hierarchical structure of the ontology consists of medical classes, instances of the medical classes, the relations (i.e., object properties in OWL) between instances, and the attributes (i.e., data properties in OWL) of instances which are derived from the guideline regimen recommendations for ICU patients. The temporal part of the ontology is the extended 4D fluent ontology which is further extended with more specific temporal classes about antibiotic dose period, interval and duration extracted from the regimen recommendations. With the assistance of the extended 4D fluent ontology, the actual time of application of antibiotic found in clinical records can be reasoned with the temporal constraints in the ontology. Thus, this antibiotic treatment guideline ontology can not only help clinicians automatically find regimen recommendations from the QUAIC antibiotic treatment guidelines and compare them with the actually used antibiotics of ICU patients, but also help clinicians check the related temporal constraints compliance issue and the temporal relations between administered antibiotics.

4.1 Guideline Patient Medical Case Analysis

As stated by QUAIC group in [43], one of major issues in the current practice of ICU clinicians is that patients in ICU often receive antibiotic therapy that is poorly chosen or is given for too many days. The purpose of the QUAIC antibiotic treatment guidelines is to provide the ICU clinicians of NSW with recommendations for the development of policies and procedures related to empirical antibiotic therapy. It aims to help the clinicians improve the quality of antibiotic treatment. The dominant part of this guideline is the antibiotic regimen recommendations provided for patients in terms of their clinical conditions. For example (Figure 6), for a febrile neutropenia patient with minor penicillin hypersensitivity, the recommended regimen is ceftazidime (2g IV 8 hourly).

In patients with minor penicillin hypersensitivity, **use**	ceftazidime	2g IV 8 hourly

Figure 6. An Example of Recommended Regimen for Febrile Neutropenia Patient

The QUAIC antibiotic treatment guidelines divide clinical conditions of ICU patients into two basic categories which are community presentation and health care associated presentation. Under the community presentation category, there are eight disease subcategories namely sepsis (uncertain focus), febrile neutropaenia, suspected fungal sepsis, community acquired pneumonia (CAP), aspiration pneumonia, suspected community acquired meningitis, trauma and urosepsis. Similarly, there are six disease subcategories under health care associated presentation which are hospital acquired pneumonia, early ventilator associated pneumonia (VAP), late VAP, intra–abdominal sepsis, biliary sepsis (cholecystitis) and acute pancreatitis. Most of the above subcategories except hospital acquired pneumonia and acute pancreatitis are further divided into more specific medical cases in terms of the combination with other clinical presentations. The following screenshot taken from the QUAIC guidelines (Figure 7) describes the community acquired pneumonia subcategory and the corresponding regimen recommendations. Six specific patient medical cases can be identified under this category. These medical cases and regimen recommendations are described in Table 2.

Regimen	Drug	Dose
1	benzylpenicillin	1.2g IV 4 hourly
	azithromycin	500mg IV daily
	gentamicin	4-6mg/kg (severe sepsis : 7mg/kg) for 1 dose, then determine dosing interval for a maximum of either 1 or 2 further doses based on renal function (see Tables 1 and 2)
	or	
2	ceftriaxone	1g IV daily
	azithromycin	500mg IV daily
	or	
In patients with immediate penicillin hypersensitivity	moxifloxacin NB1 and NB2	400mg IV daily
	azithromycin	500mg IV daily
	Special circumstances	
In any patient with suspected staphylococcal pneumonia from Gram stain of sputum, clinical picture, radiographic appearance and/or initial blood culture result) **add**	vancomycin	1.5g IV 12 hourly (adjust initial dosage for renal function, and subsequent doses to achieve therapeutic range (Antibiotic Guidelines 14 Table 26 p365). For information on continuous infusion, see Antibiotic Guidelines 14 p365.
In any patient with severe pneumonia with a clinical presentation consistent with severe influenza, during a period where Influenza A is known to be circulating	consider neuramindase inhibitor (oseltamivir, zanamivir)	150mg via nasogastric tube twice daily

Figure 7. Community Acquired Pneumonia and the Regimens

1. Patient who has community acquired pneumonia, but has not had severe sepsis and penicillin hypersensitivity	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly)
2. Patient who has community acquired pneumonia and severe sepsis, but has not had penicillin hypersensitivity	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly)
3. Patient who has community acquired pneumonia and suspected staphylococcal pneumonia, but has not had penicillin hypersensitivity and severe sepsis	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly); Vancomycin (150 mg IV 12 hourly)
4. Patient who has community acquired pneumonia, suspected staphylococcal pneumonia and severe sepsis, but has not had penicillin hypersensitivity	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly); Vancomycin (150 mg IV 12 hourly)
5. Patient who has community acquired pneumonia and immediate penicillin hypersensitivity	Azithromycin (500 mg IV 24 hourly); Moxifloxacin (400 mg IV 24 hourly)
6. Patient who has severe community acquired pneumonia and severe influenza that is in the period when influenza A virus is circulating	Neuramindase Inhibitor (Oseltamivir **OR** Zanamivir) (150 mg nasogastric tube 12 hourly)

Table 2. Identified Medical Cases and Regimens under the Community Acquired Pneumonia Category

The QUAIC guidelines provide antibiotic regimen recommendations for each patient's medical case with two exceptions. One of the two exceptions is for a patient who has sepsis and hypersensitivities where neither recommended antibiotic regimen nor general medical recommendation is available. The guideline only recommends clinicians to refer to another guideline. Another one is for patient who has hospital acquired pneumonia where only general medical recommendations are available. Most drugs in these regimen recommendations are antibiotics other than three antiviral drugs (i.e., acyclovir, oseltamivir and zanamivir). Moreover, almost each regimen recommendation has the repetitive temporal constraint as a part of dosage instruction. In the previous example of febrile neutropenia patient with minor penicillin hypersensitivity, the "8 hourly" is a periodical interval constraint for ceftazidime dosage. Some temporal constraints in the dosage instructions have both periodical intervals and duration. For example, for orthopaedics trauma patients with fracture size less than 1 cm (Gustillo Type I), use 2 cefazolin (2g IV 8 hourly) or vancomycin (1.5g IV 12 hourly) 24 hours after wound closure or 2 days for open wound (see medical cases 1 and 2 in trauma category in Appendix 1).

A patient medical case classification list (Appendix 1) is developed from the QUAIC guidelines for describing the patient clinical conditions and the recommended regimens. There are 66 medical cases in the list and all of them are verified by an ICU medical expert in our research group. From these medical cases, medical concepts, relations and attributes for describing patient clinical conditions and antibiotic regimen recommendations are extracted and are organised in the antibiotic treatment guideline ontology. For example, under the aspiration pneumonia subcategory, medical case 1 (Appendix 1) is about patient who has aspiration pneumonia, but has not had penicillin hypersensitivity and pseudomonal pneumonia; and, the recommended antibiotic regimen is metronidazole (500 mg IV 12 hourly) and benzylpenicillin (120 mg IV 4 hourly). From this case, ontology concepts of about diseases (i.e., aspiration pneumonia, penicillin hypersensitivity and pseudomonal pneumonia), drugs (i.e., metronidazole and benzylpenicillin), and attributes of medications (i.e., dose agency, dose amount, dose interval, route of administration) can be extracted.

4.2 The Structure of the Antibiotic Treatment Guideline Ontology

The extracted classes are organised into two parts in the antibiotic treatment guideline ontology. The domain ontology is used to represent the medical knowledge contained in the regimen recommendations and the extended 4D fluent ontology is used to represent the temporal knowledge in the recommendations.

4.2.1 The Domain Ontology for Modelling Medical Knowledge in the Guideline

The domain ontology contains all medical classes, relations and attributes which are extracted from the regimen recommendations and listed in Appendix 2, 3, 4, 5 and 6. In the domain ontology, four superclasses which are the four SNOMED CT top level concepts namely “Clinical Finding”, “Drug”, “Procedure” and “Social Context” are defined to classify the medical knowledge contained in the regimen recommendations. Medical knowledge about ICU patient such as disease, administered antibiotics and recommended regimens is organised as their subclasses. Figure 8 is a part of disease classes and Figure 9 is a part of drug classes extracted from the medical cases.

Figure 8. A Part of the Ontology Structure about Disease

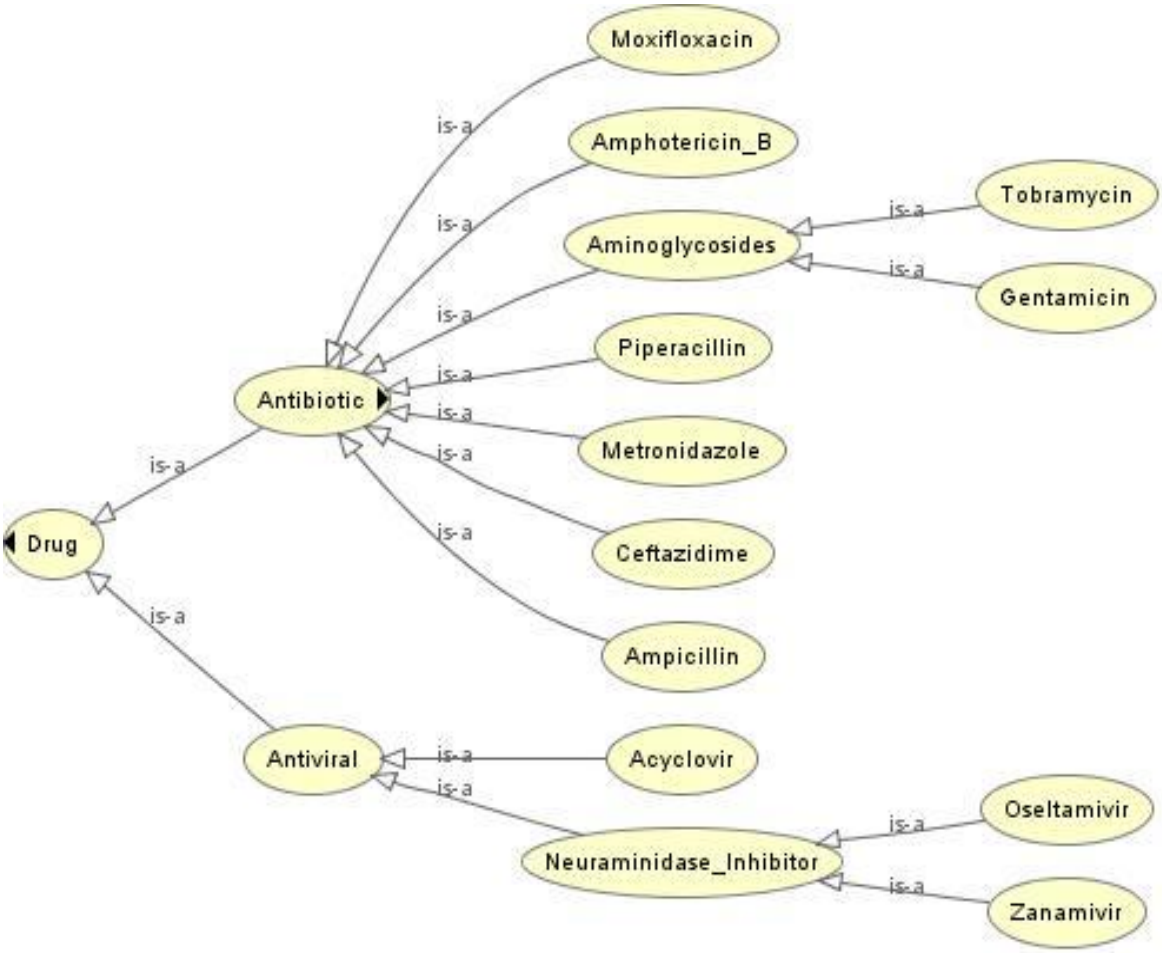


Figure 9. A Part of the Ontology Structure about Drug

The relations between instances of these medical classes in the ontology are represented by OWL object properties. The attributes of instances are represented by OWL data properties. The following example in Figure 10 is about the relation between patient Lucy and her clinical conditions, and the attributes of recommended regimens.

Patient Lucy has sepsis and shock. The recommended regimen is medication 1 (flucloxacillin 200 mg IV 6 hourly).

Figure 10. Clinical Conditions and Regimen of Patient Lucy

To represent the relation between Lucy and her diseases, an object property namely "present" is defined in the ontology. To represent the attributes of the medication 1, four data properties namely "dose_agent", "dose_amount", "interval" and "route_of_administration" are defined in the ontology. The relation and attributes are visualised in Figure 11 and Figure 12 respectively.

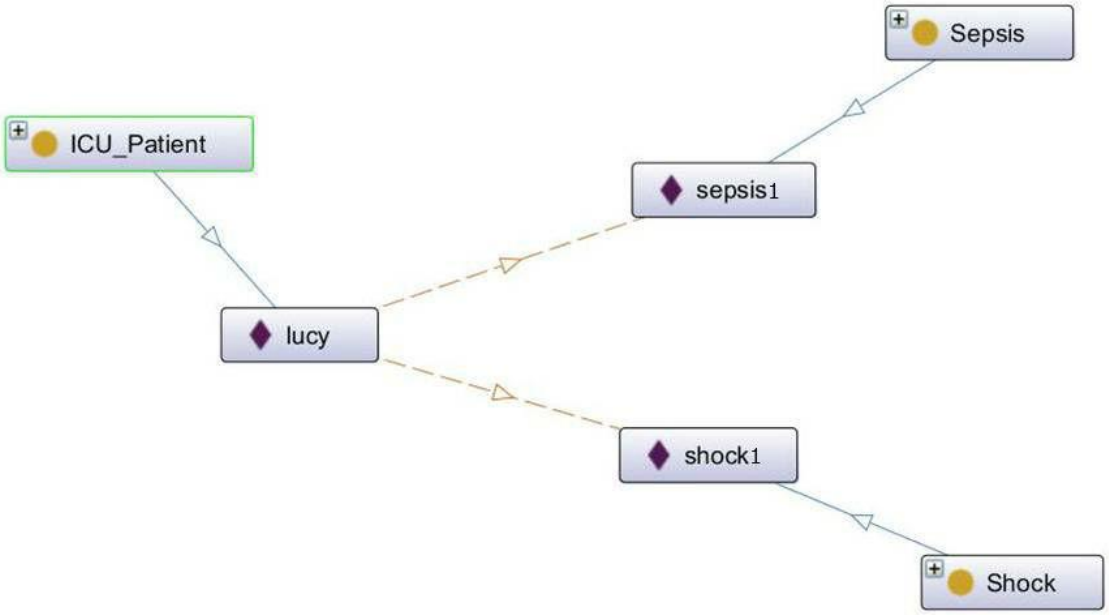


Figure 11. The Relation between Patient Lucy and Her Disease

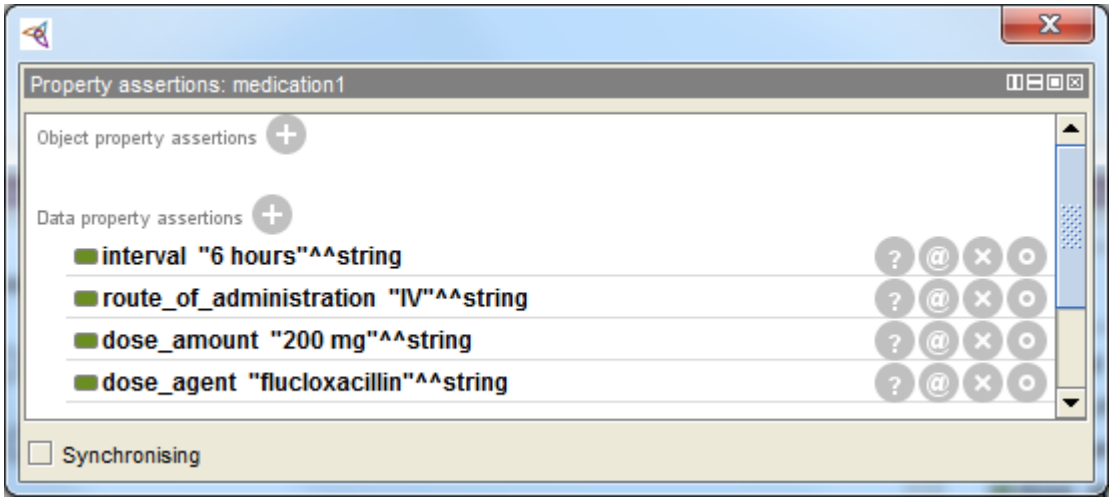


Figure 12. Attributes of Medication1 for Patient Lucy

The corresponding OWL syntax for the classes, instances, relations and attributes in the ontology is represented in the RDF/XML serialisation format since it is only the one that all OWL ontology tools can parse. The relations and attributes in the previous example (Figure 10) are shown in the RDF/XML serialisation format below.

```
<rdf:RDF xmlns="http://www.usyd.edu.au/hitru/antibiotics#"
    xml:base="http://www.usyd.edu.au/hitru/antibiotics"
    xmlns:antibiotics="http://www.usyd.edu.au/hitru/antibiotics#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
   <owl:Ontology rdf:about="http://www.usyd.edu.au/hitru/antibiotics"/>
   <owl:NamedIndividual rdf:about="&antibiotics;sepsis">
      <rdf:type rdf:resource="&antibiotics;Sepsis"/>
   </owl:NamedIndividual>
   <owl:NamedIndividual rdf:about="&antibiotics;shock1">
      <rdf:type rdf:resource="&antibiotics;Shock"/>
   </owl:NamedIndividual>
   <owl:NamedIndividual rdf:about="&antibiotics;lucy">
      <rdf:type rdf:resource="&antibiotics;ICU_Patient"/>
      <present rdf:resource="&antibiotics;sepsis1"/>
     <present rdf:resource="&antibiotics;shock1"/>
   </owl:NamedIndividual>
   <owl:NamedIndividual rdf:about="&antibiotics;medication1">
      <rdf:type rdf:resource="&antibiotics;Recommended_Regimen"/>
      <dose_amount rdf:datatype="&xsd;string">200 mg</dose_amount>
      <interval rdf:datatype="&xsd;string">6 hours</interval>
      <route_of_administration rdf:datatype="&xsd;string">IV</route_of_administration>
      <dose_agent rdf:datatype="&xsd;string">flucloxacillin </dose_agent>
   </owl:NamedIndividual>
</rdf:RDF>
```

4.2.2 The Extended 4D Fluent Ontology for Modelling Temporal Knowledge in the Guidelines

In order to implement the temporal reasoning related to dose interval and dose duration compliance checking in antibiotic administration, more specific temporal classes are added to the extended 4D fluent ontology (Appendix 7). These classes (Figure 13) are explained below.

- Two classes namely "ICUPatient_TimeSlice" and "AdministeredRegimen_TimeSlice" are created under "Time_Slice" class to hold the temporal parts of ICU patients and administered antibiotics.
- Three classes "Dose_Start_Time", "Dose_Following_Time" and "Dose_End_Time" are created under "Time_Instant" class hierarchy to hold each dose time of administered antibiotics.
- A "Dose_Interval" class is created under "Time_Interval" class to hold each dose interval of administered antibiotics.
- Four classes namely "Quantitative_Dose_Period", "Qualitative_Dose_Period", "Left_Close_Dose_Period" and "Right_Close_Dose_Period" are created under "Time_Period" class hierarchy. The "Quantitative_Dose_Period" class is used to hold dose periods where both the values of dose start time and dose end time are known, whereas the "Qualitative_Dose_Period" class is used to hold dose periods where both the values of dose start time and dose end time are not known. The "Left_Close_Dose_Period" class is used to hold the dose periods where the value of dose end time is unknown; and, the "Right_Close_Dose_Period" is used to hold the dose periods where the value of dose start time is unknown. Quantitative dose periods are often found in clinical database whereas qualitative dose period and semi-quantitative dose periods are often found in clinical notes.

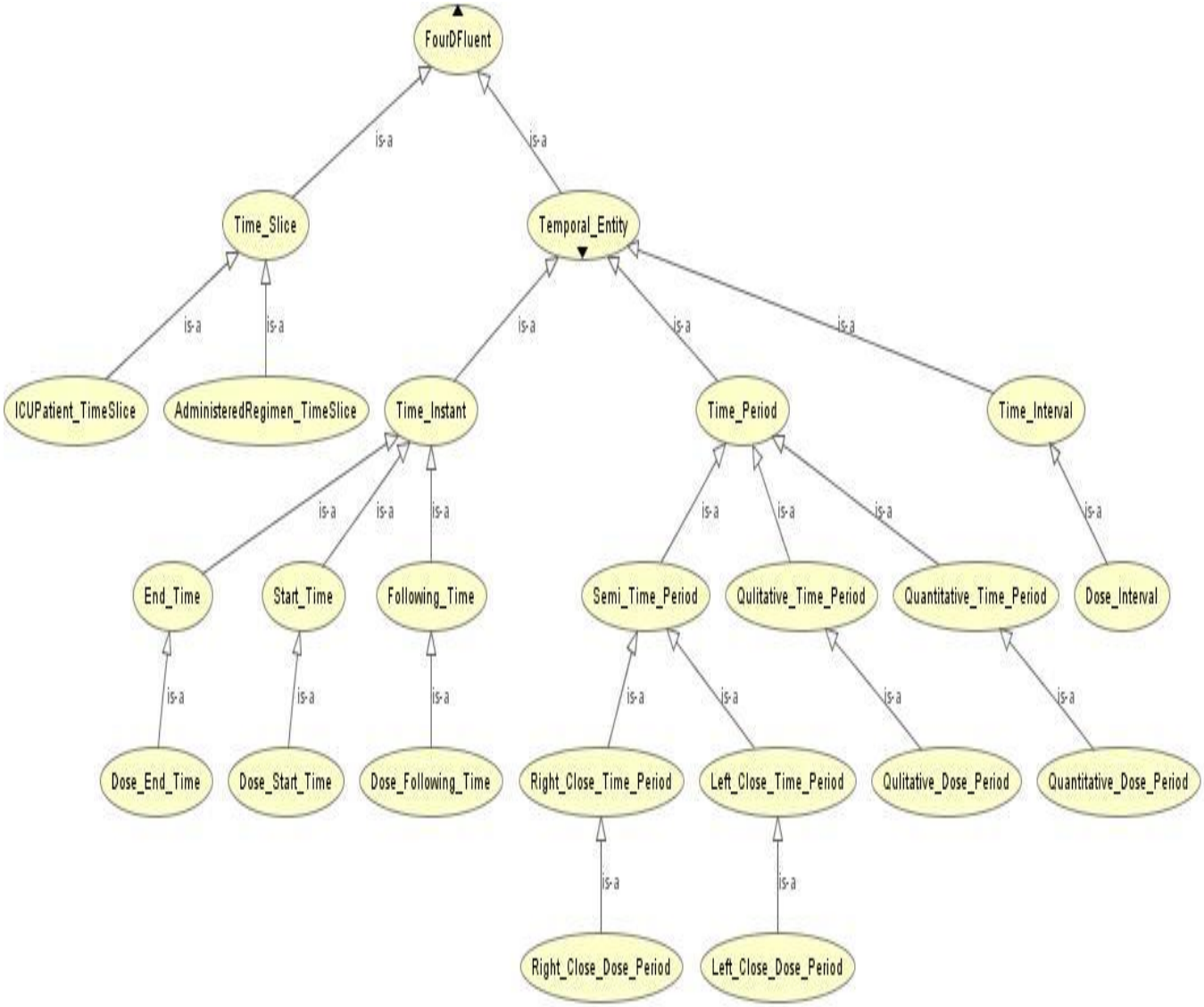


Figure 13. The Further Extended 4D Fluent Ontology

These temporal classes defined in the extended 4D fluent ontology enable the modelling of the relation between ICU patients and administered antibiotics that holds in a particular time instant or time period, the relation between dose time and dose interval, and the relation between dose time and dose period of administered antibiotics. In the previous example (Figure 10), Lucy was administered flucloxacillin in a time period that started from time t1, followed by t2, t3, t4, and ended at t5. In this example, there are five dose time points or instants, four dose intervals between t1 and t2, t2 and t3, t3 and t4, and one dose period between t1 and t5. The object property “administered_with” is defined in the

ontology to model the relation between Lucy and flucloxacillin that holds in a time instant based on the 4D fluent representation method (Figure 14); whereas another two object properties namely “open_instant” and “close_instant” are defined in the ontology to model the relation between dose time and dose interval and the relation between dose time and dose period (Figure 15). The object property “has_temporal_part” is used to connect the patient Lucy or the antibiotic flucloxacillin with its time slices; whereas the object property “has_temporal_entity” is used to connect these time slices with temporal entities such as dose time.

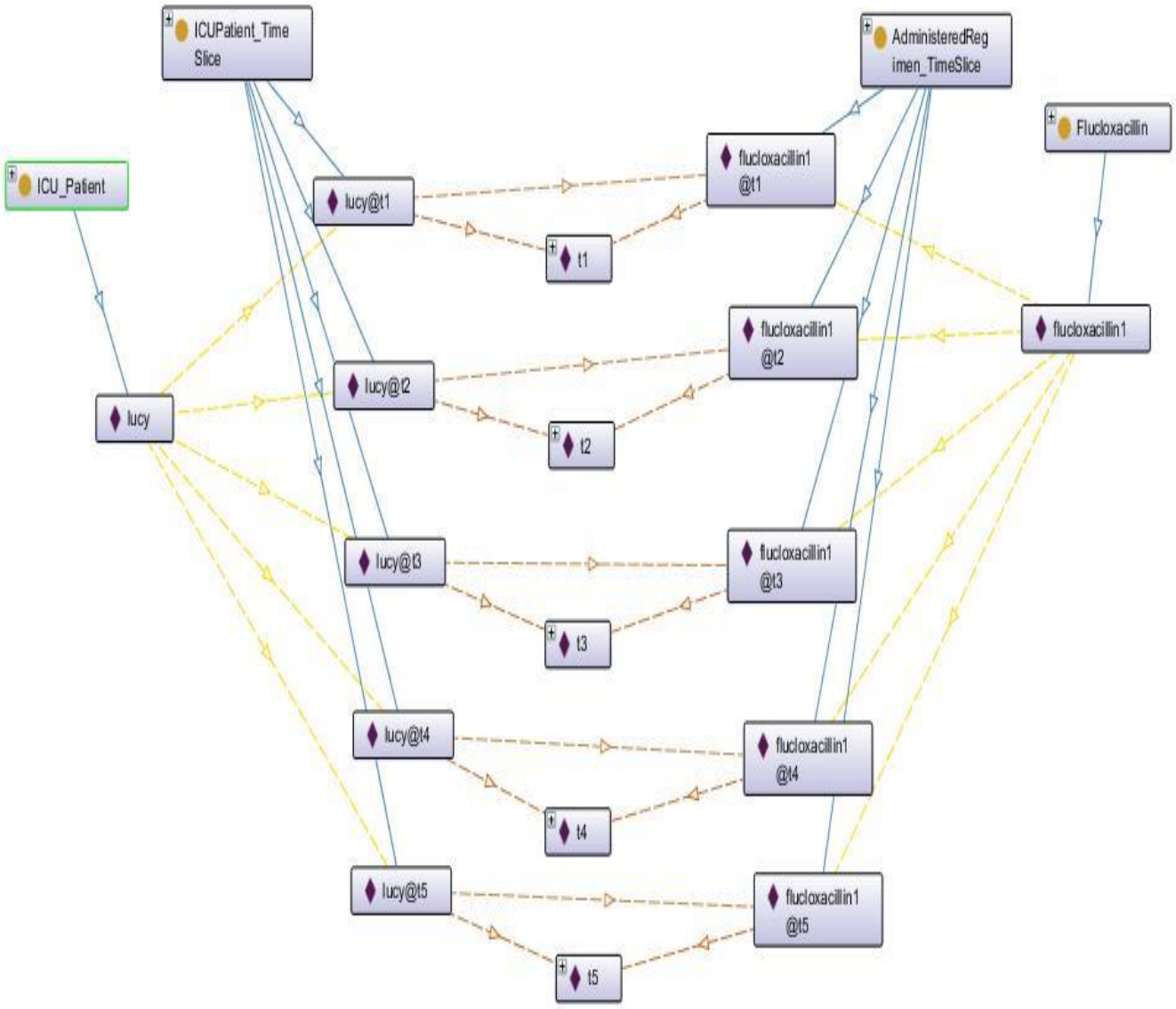


Figure 14. The Antibiotic Administration Relation Holding in Time Points

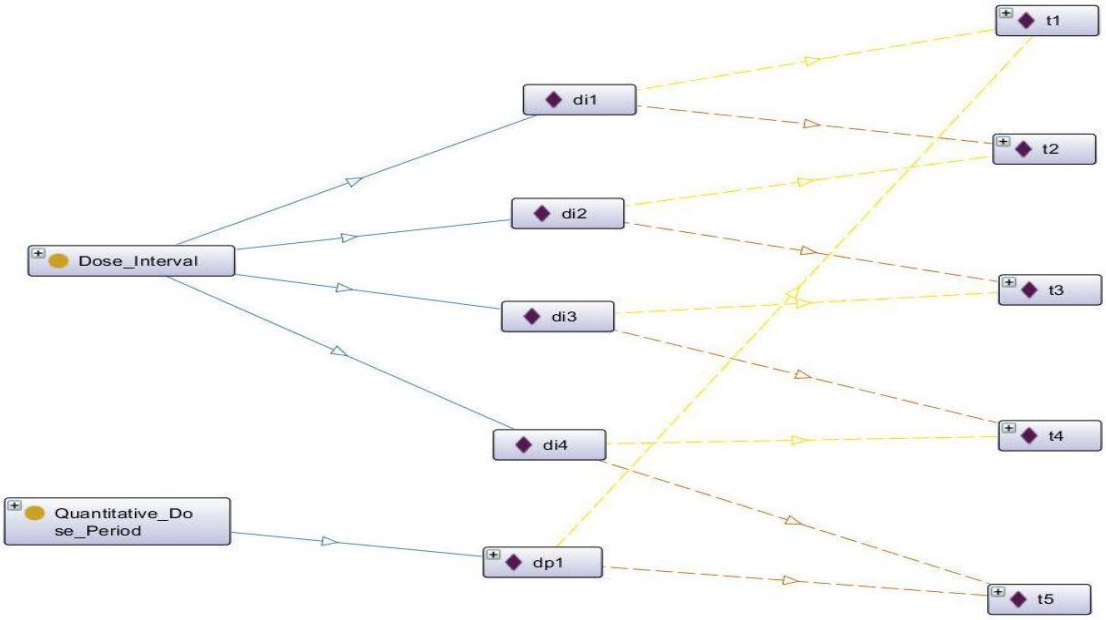


Figure 15. The Relations between Dose Time, Dose Interval and Dose Period

The repetitive temporal constraints contained in the guideline regimen recommendations can be added into the ontology under these classes. Seven dose intervals with different length in hours contained in the recommendations are added into the ontology under the "Repetitive_Temporal_Constraint" class hierarchy (Figure 16). Some of these dose intervals in recommendations also have dose duration constraints. These dose duration constraints are added into the ontology under the "Time_Duration" class hierarchy (Figure 17). During the reasoning process of the ontology, the actual time of antibiotic dose found in clinical records will be used to reason with these temporal constraints for dose interval and dose duration compliance checking.

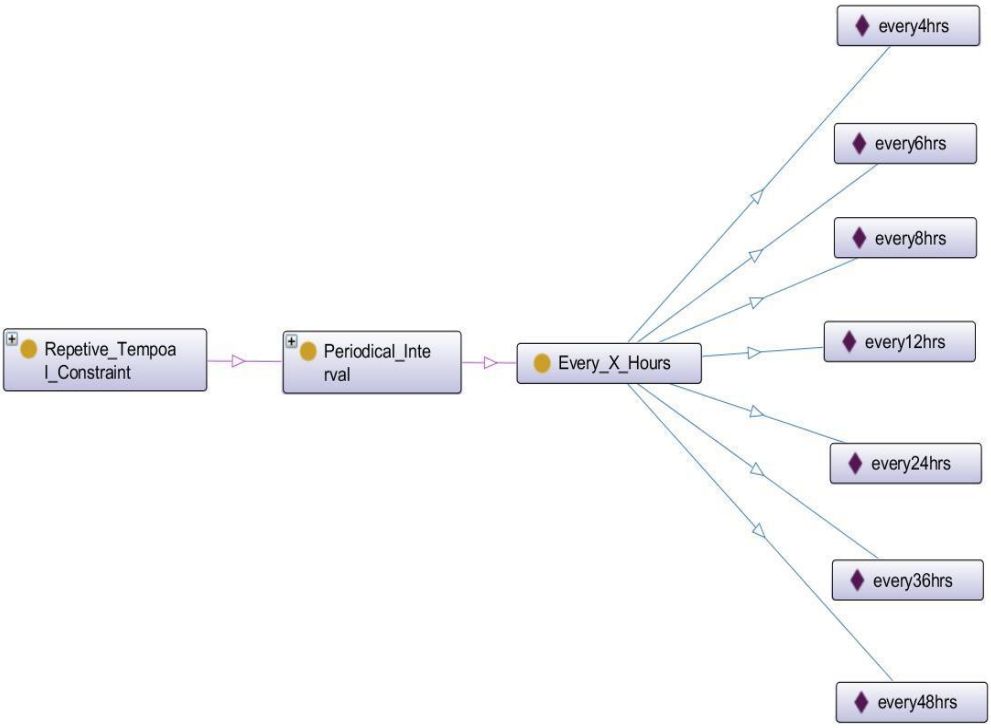


Figure 16. Dose Intervals in the Guideline Regimen Recommendations

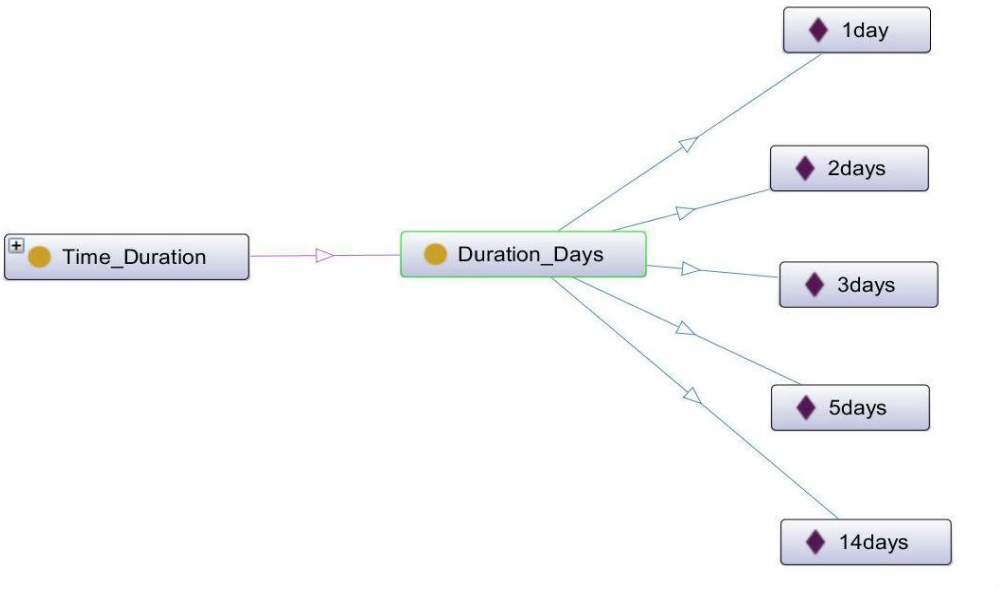


Figure 17. Dose Durations in the Guideline Regimen Recommendations

The extended 4D fluent ontology also enables modelling the temporal relations between administered antibiotic if the temporal relations between these drugs cannot be determined by just comparing the values of their administered time. Since some of those dose periods are qualitative, the values of dose start time and dose end time are not known. Therefore, the temporal relations between the administered antibiotics cannot be determined by comparing the values of administered time. In the previous example (Figure 10), assume that Lucy was administered flucloxacillin in the period dp1 (from t1 to t5) and was then administered another antibiotic vancomycin in the period dp2 (from t6 to t8) after flucloxacillin. There is a "before" temporal relation between these two clinical events. The temporal relation "before" between the two events is actually the temporal relation between the dose period of flucloxacillin and the dose period of vancomycin. Based on the 4D fluent representation method, the modelling of the "before" relation can be illustrated in Figure 18. With the quantitative, qualitative and semi-quantitative temporal information about administered antibiotics populated into the extended 4D fluent ontology, the temporal relation reasoning rules defined in the ontology can infer various temporal relations between these drugs.

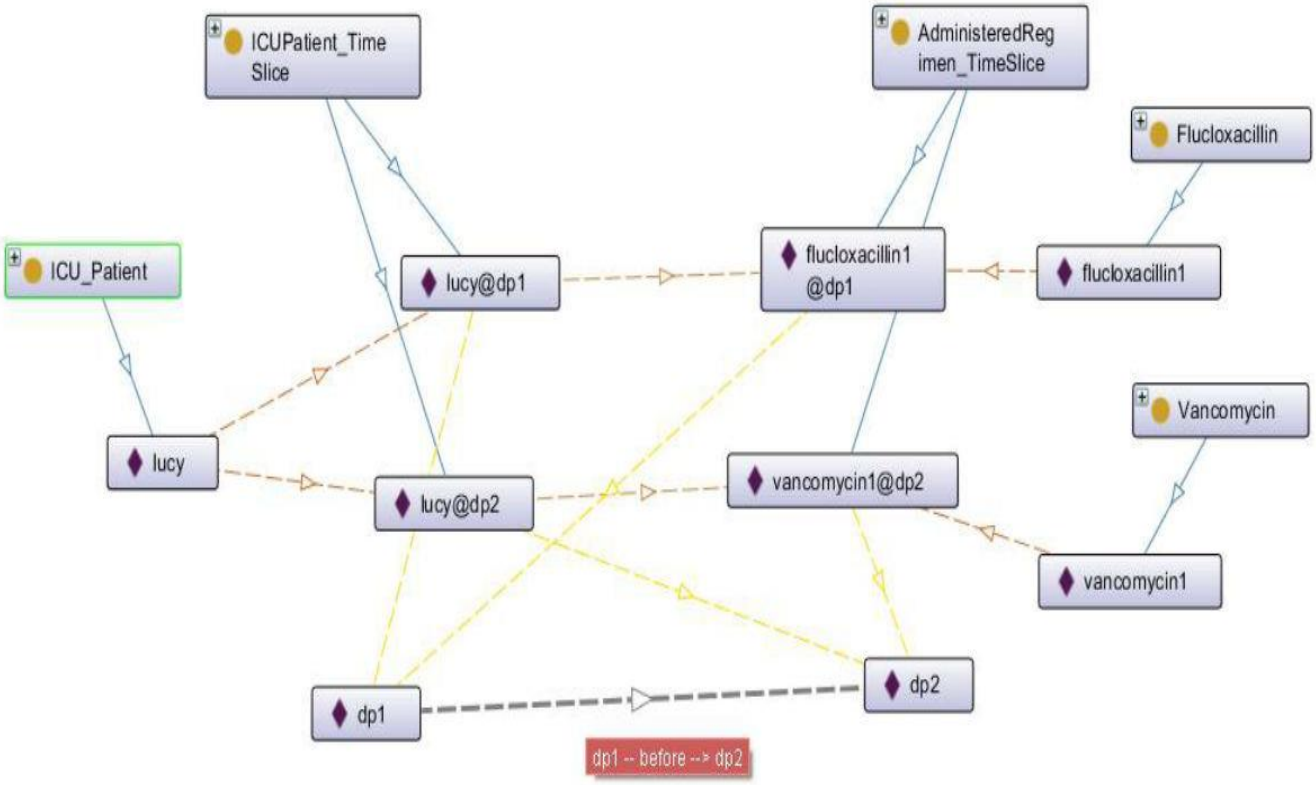


Figure 18. The "Before" Relation between Administered Drug Flucloxacillin and Vancomycin

4.3 Terminology Mapping between the Guideline Ontology and SNOMED CT

Clinical guidelines from different hospitals often use different terminologies to describe the same medical knowledge. This situation creates an obstacle for the communication between clinicians. In order to facilitate the medical terminology interoperability between clinical guidelines and the international clinical terminology standard SNOMED CT, it is necessary to map the medical concepts in the antibiotic ontology into SNOMED CT concepts. There are two types of medical concepts in SNOMED CT which are the pre-coordinated concept and post-coordinated concept. The pre-coordinated concept is the representation of a clinical meaning using a single concept identifier whereas the post-coordinated concept is the representation of a clinical meaning using a combination of two or more concept identifiers [44]. However, according to an empirical study of six international preoperative assessment clinical guidelines, Ahmadian et al. [45] found that SNOMED CT is not able to cover and represent all medical concepts in these guidelines. Among 133 extracted terms in their study, 80% of them (i.e., 107 terms) can be covered by SNOMED CT. Moreover, 68% of these 107 terms can be completely represented by SNOMED CT pre-coordinated concepts and 19% of them can be mapped into the post-coordinated concepts.

Generally speaking, the simpler a term is, the more likely it can be mapped into the pre-coordinated concepts; the more complex a term is, the more likely it can be mapped into the post-coordinated concepts. That is because more complex terms involve the combination of different terms and the relations between terms. Thus, it needs extra SNOMED CT pre-coordinated concepts, attributes or qualifiers for the mapping in the form of the compositional grammar of SNOMED CT.

For example, the medical concept “a procedure that replaces a left hip with insertion of a prosthesis” [44] is a complex concept. In order to map the concept into a SNOMED CT post-coordinated concept in the ontology, a subset of SNOMED CT has to be imported into the ontology. Then, the concept can be mapped to a post-coordinated concept in the form of the compositional grammar using Manchester Syntax (an OWL syntax serialisation format) in Protégé. For the example above, it could be written in the following Manchester Syntax as shown below.

Prosthetic arthroplasty of the hip (procedure) and

((Procedure site (attribute) some Hip joint structure (body structure)) and

(Laterality (attribute) some Left (qualifier value))) and

(RoleGroup some

(Direct device (attribute) some (Total hip replacement prosthesis (physical object))) and

('Method (attribute)' some 'Insertion-action (qualifier value)'))

However, mapping complex concepts into SNOMED CT concepts will make the guideline ontology very complicated. As a result, the ontology maintenance will become more difficult. To simplify the antibiotic treatment guideline ontology, the concept mapping is only limited within the pre-coordinated concepts of SNOMED CT using two user defined annotation properties which are “sctCode” and “sctName” in Protégé. For example, the concept of “biliary obstruction” in a regimen recommendation is mapped to the SNOMED CT concept “Obstruction of biliary stent (disorder)” as show in Figure 19.

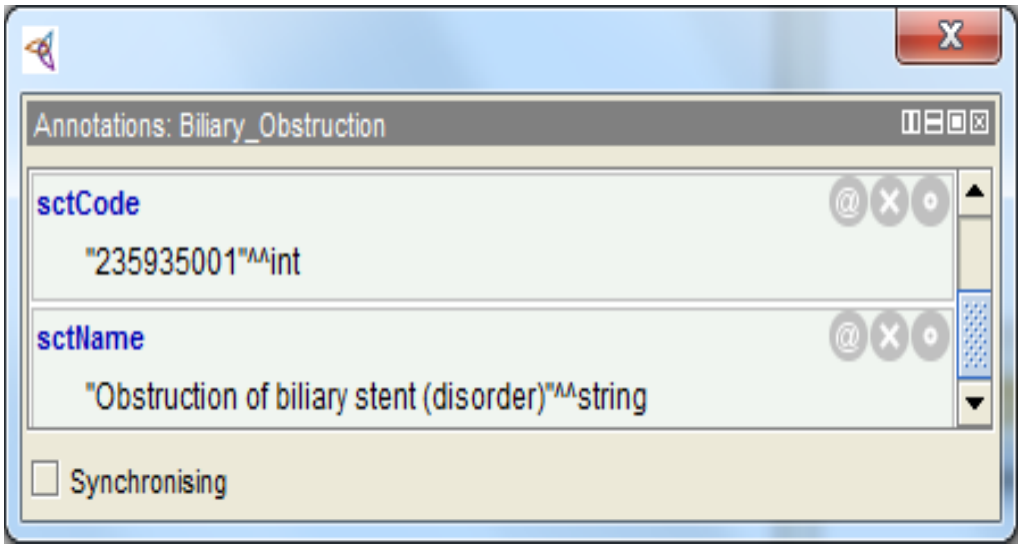


Figure 19. A Mapping Example Using Annotation Properties in Protégé 4.1

In summary, the clinical knowledge and temporal knowledge contained in the antibiotic treatment guidelines is analysed and modelled in the antibiotic treatment guideline ontology. Clinical and temporal knowledge about ICU patients can be precisely represented with these medical and temporal classes, relations, and attributes defined in the guideline

domain ontology and the extended 4D fluent ontology. This makes it possible to implement knowledge reasoning to infer the implicit knowledge in the antibiotic treatment guideline knowledge base. By leveraging the reasoning rules and the input of patient data in the reasoning process, the guideline ontology can answer some important clinical questions related to the guideline.

Chapter 5 Clinical and Temporal Knowledge Reasoning in the Antibiotic Treatment Guideline Ontology

From the perspective of clinical practice with regard to the QUAIC antibiotic treatment guidelines, clinicians often consult the guidelines with the following questions.

- What are the recommended antibiotic regimens for a patient if the patient has the clinical presentation described in that guideline such as sepsis and pneumonia?
- What are the administered antibiotics for a patient who has recommendations and are these drugs different than the ones recommended by that guideline?
- For the patient who has regimen recommendations, what are the actual dose intervals and dose durations of the administered antibiotics and do they follow the recommended temporal constraints?
- What are the temporal relations between administered antibiotics for that patient?
- Is there any inconsistent temporal relation between administered antibiotics which might occur in the antibiotic treatment guideline ontology?

Answering these questions in the QUAIC antibiotic treatment guideline ontology can help clinicians research and review their antibiotic administration practice on ICU patients. These questions not only involve clinical knowledge about recommendations and administered drugs, but also involve temporal knowledge which is important for antibiotic administration. All of these questions except the first one involve some temporal knowledge since the related queries for answering these questions contain some temporal graph pattern matching based on the extended 4D fluent representation method. In order to answer these questions, a rule-based knowledge reasoning system is developed in Oracle RDF Semantic Graph (a native triple store for ontology in Oracle 12c) and Jena (a Java API for ontology). The reasoning system has two parts which are the clinical knowledge reasoning part for finding regimen recommendations and administered antibiotics, and the temporal knowledge reasoning part for checking dose interval, dose duration and temporal relations of administered antibiotics.

5.1 Clinical Knowledge Reasoning for Finding Regimen Recommendations

5.1.1 Analysis of the Relation between Patient Medical Cases and Reasoning Rules

One of most important features in ontology-based systems is the reasoning feature which is not supported in the traditional relational database systems. OWL 2 contains a set of axioms (statements about what is true in the domain) for inferencing the relations among classes, object properties, data properties and individuals etcetera. Many tools such as Pellet, FaCT++, HermiT, RacerPro and Oracle RDF Semantic Graph etcetera provide support in OWL reasoning based on these axioms. For example, the subclass axiom allows the inference of the subclass relation between class A and class C if a class A is a subclass of B and B is a subclass of C. The inverse object property axiom allows a new assertion "Person A is the father of person B" to be inferred from the assertion "Person B has a father who is person A" since the inverse of the object property "hasFather" is the property "isFatherOf". Moreover, the complex object subproperty axiom involving an object property chain allows simple user defined reasoning rules in OWL 2. For example, if X has a mother who is Y and Y has a sister who is Z, then X has an aunt who is Z. By leveraging that axiom, a rule could be defined as SubObjectPropertyOf(ObjectPropertyChain(*:hasMother :hasSister*) *:hasAunt*) using the Functional Syntax in OWL 2 [13].

In addition to these powerful reasoning features of OWL 2, many ontology rule languages such as SWRL, Jena rules and Oracle user-defined rules are developed to extend the existing reasoning capability of OWL 2. Under these rule languages, ontology developers can define their own customised rules which are not representable or very cumbersome to be represented in OWL 2. These rules usually involve class expression and numeric or time computation. For example, the following simple rule has a class expression "Man(?x)" in its rule body.

Man(?x) ∧ hasChild(?x, ?y) -> fatherOf(?x, ?y)

This rule can be easily realised in rule languages; but, it cannot be directly expressed in an object subproperty axiom involving property chain in OWL 2. It needs to use a kind of rolification workaround technique to solve it. That is to say it needs to convert the class

"Man" into an object property such as "pMan" in the following auxiliary axiom which states the class Man is a thing that has a pMan relation to itself [46].

$$Man \equiv \exists P Man.Self$$

Then, an object subproperty axiom can be defined for the previous rule in the following manner using OWL Functional Syntax.

SubObjectPropertyOf(ObjectPropertyChain(*:pMan :hasChild*) *:fatherOf*)

As to the rules involving numeric or time computation such as addition, subtraction, multiplication and division, there is no built-in functions in OWL 2 to implement the related computation, but it can be achieved in SWRL and Oracle RDF Semantic Graph.

Reasoning rules in ontology rule languages generally have this following logical form in which if the premises or antecedents are true, the consequent is also true.

$p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q$

The consequent q is said to be semantically entailed by its premises; and, this form is said to be a logically valid argument. Deductive reasoning is just an approach to find and check this kind of valid arguments in the domain of discourse [47].

In the patient medical case list (Appendix 1) which is analysed in Chapter 4, each case basically contains two parts: patient clinical conditions and the recommended regimens. In terms of rules, the first part can be viewed as the antecedent of the rule whereas the second part can be viewed as the consequent of the rule. Therefore, these patient medical cases can be formalised in rules. For example, the following medical case describes the clinical conditions of a febrile neutropenia patient and the recommended regimen.

> If a patient has febrile neutropenia patient and minor penicillin hypersensitivity, the recommended regimen is medication 2 (ceftazidime 200 mg IV 8 hourly)

This medical case implies a rule which has the following logic form.

> ICU_Patient(?x) ∧ Febrile_Neutropenia (?y) ∧ Minor_Penicillin_Hypersensitivity (?z) ∧ present(?x, ?y) ∧ present(?x, ?z) -> has_recommendation(?x, medication2)

From this rule, a conclusion (i.e., the recommended regimen) can be deducted from its antecedents (i.e., if the patient has the diseases described in this case). The following Oracle user defined rule could be written to represent that medical case.

```
INSERT INTO mdsys.semr_rulebase1 VALUES('rule1',
'(?x rdf:type :ICU_Patient) (?y rdf:type :Febrile_Neutropenia) (?x :present ?y)
 (?z rdf:type :Minor_Penicillin_Hypersensitivity) (?x :present ?z)',
 null,
'(?x :has_recommendation :medication2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

This Oracle rule has a rule name, an IF side pattern containing the antecedents, a filter condition, a THEN side pattern containing the consequents and one or more namespaces represented by the URI. The filter condition is used to further restrict the graph matching in the IF side pattern. The null value denotes there is no filter condition to be applied [48].

In order to infer a conclusion from that rule, this rule needs to be added into a user defined rulebase such as “rulebase1”. Then, an inference entailment needs to be created to store pre-computed triples which are inferred from applying one or more rulebases to a semantic model of the antibiotic treatment guideline ontology. For the previous rule, the corresponding entailment is shown below.

```
BEGIN
SEM_APIS.CREATE_ENTAILMENT (
'rix1',
SEM_Models ('antibiotics'),
SEM_Rulebases ('owl2rl', 'rulebase1'),
SEM_APIS.REACH_CLOSURE,
NULL,
'USER_RULES=T');
END;
```

This entailment has an entailment name, a semantic model name, two rulebases (one is the Oracle built-in OWL2 RL rulebase for the inference of OWL 2 axioms; another one is for the user defined rules), and other parameters to restrict the reasoning process [48].

There are total 66 patient medical cases listed in Appendix 1. However, the relation between these medical cases and rules is not a simple one-to-one mapping relation. That is

because of some limitations in ontology rule languages such as SWRL, Jena rules and Oracle user defined rules.

Generally speaking, only positive conjunctions of atomic formulas are supported in the rule body. If the rule body contains the disjunction of atomic formulas, it should be placed in different rules. For example, rules in the following logic form cannot be directly represented in Oracle user defined rules since its rule body is a disjunction of n atoms.

$p_1 \vee p_2 \vee \ldots \vee p_n \rightarrow q$

It needs to be converted to the following set of n rules to achieve the desired effect.

$p_1 \rightarrow q$

$p_2 \rightarrow q$

…

$p_n \rightarrow q$

Moreover, the consequent of a rule should be an atomic head. It means the rule head cannot be in the form of disjunction or conjunction of atomic formulas. For example, the following logical form which involves a disjunctive rule head cannot be represented in Oracle user defined rules.

$p_1 \wedge p_2 \ldots \wedge p_n \rightarrow q_1 \vee q_2 \ldots \vee q_m$

However, if the rule head is a conjunction of two atoms as shown below, the atoms should be placed in different rules to achieve the desired effect.

$p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q_1 \wedge q_2 \ldots \wedge q_m$

Thus, it needs to be converted to the following set of rules.

$p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q_1$

$p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q_2$

…

$p_1 \wedge p_2 \wedge \ldots \wedge p_n \rightarrow q_m$

Furthermore, negation as failure is not supported as well due to the monotonic nature of ontology rule languages. For example, the following rule form containing the negation of an atomic formula in the rule body cannot be represented in Oracle user-defined rules.

$p_1 \wedge p_2 \ldots \wedge p_n \wedge \neg r \rightarrow q$

In the 66 patient medical cases, many of them involve the conjunctive rule head, disjunctive rule head and negation as failure as discussed previously. The following medical case and its logical form involve a conjunctive rule head.

If a patient has febrile neutropenia, shock and minor penicillin hypersensitivity, the recommended regimens are medication 1 (ceftazidime 200 mg IV 8 hourly) and medication2 (vancomycin 150 mg IV 12 hourly).

ICU_Patient(?x) ∧ Febrile_Neutropenia (?y) ∧ Minor_Penicillin_Hypersensitivity (?z) ∧ Shock(?m) ∧ present(?x, ?y) ∧ present(?x, ?z) ∧ present(?x, ?m) -> has_recommendation(?x, medication1) ∧ has_recommendation(?x, medication2)

The recommended regimen in this medical case is a conjunction of two medications. Therefore, this medical case can be represented in the following two user-defined rules.

```
INSERT INTO mdsys.semr_rulebase1 VALUES('rule1',
'(?x rdf:type :ICU_Patient) (?x :present ?y) (?y rdf:type :Febrile_Neutropenia)
 (?x :present ?z) (?z rdf:type :Minor_Penicillin_Hypersensitivity)',
 null,
'(?p :has_recommendation :medication1)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

```
INSERT INTO mdsys.semr_rulebase1 VALUES('rule2',
'(?x rdf:type :ICU_Patient) (?x :present ?y) (?y rdf:type :Febrile_Neutropenia)
 (?x :present ?m) (?m rdf:type :Shock)',
 null,
'(?p :has_recommendation :medication2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

This medical case and its logical form below involve a disjunctive rule head.

If a patient has suspected community acquired meningitis, the recommended regimen is medication 1 (ceftriaxone 4g IV 24 hourly) or medication 2 (cefotaxime 2g IV 6 hourly)

ICU_Patient(?x) ∧ Suspected_Community_Acquired_Meningitis(?y) ∧ present(?x, ?y) -> has_recommendation(?x, medication1) ∨ has_recommendation(?x, medication2)

It can be achieved in the following workaround by creating a new medication instance "medication1_or_medication2".

```
INSERT INTO mdsys.semr_rulebase1 VALUES ('rule3',
'(?x rdf:type :ICU_Patient) (?x :present ?y)
(?y rdf:type :Suspected_Community_Acquired_Meningitis)',
 null,
'(?p :hasRecommendation :medication1_or_medication2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

However, for negation as failure in the following medical case and its logical form, there is no built-in function for negation such as “NOT EXISTS” in SPARQL (SPARQL Protocol and RDF Query Language) can be used in Oracle user-defined rules to define a negation rule.

If a patient has aspiration pneumonia, but has not had penicillin hypersensitivity and pseudomonal pneumonia, the recommended regimen is medication 3 (metronidazole 500 mg IV 12 hourly) and medication 4 (benzylpenicillin 120 mg IV 4 hourly)

ICU_Patient(?x) ∧ Aspiration_Pneumonia(?y) ∧ Penicillin_Hypersensitivity(?z) ∧ Pseudomonal_Pneumonia(?m) ∧ present(?x, ?y) ∧ ¬ present(?x, ?z) ∧ ¬ present(?x, ?m) -> has_recommendation(?x, medication3) ∧ has_recommendation(?x, medication4)

In order to deal with the negation issue, the inference extension architecture provided by Oracle 12c is leveraged to achieve the intended result. The inference extension architecture, as the complement of user defined rules, enables developers to create a user defined inference function such that it can add user defined inferencing to the pre-supplied inferencing support [48]. To create a negation rule for the above medical case, the “NOT EXISTS” built-in function can be used in a user defined inference function (Appendix 9) to infer the recommended regimen. To infer the regimen recommendations for the medical case involving negation, the function needs to be called in an inference entailment. The core part of the inference function shown below is that it leverages three SQL select queries to find all patients who have aspiration pneumonia, but do not have penicillin hypersensitivity and pseudomonal pneumonia, and insert all eligible patients, the recommended medications into the semantic model of the ontology in the form of subject-predicate-object format. For other medical cases involving negation, the functions are implemented in the similar way.

```
-- extract the ID of patients, diseases and the “present” property to
-- find all patients who have aspiration pneumonia,
-- but do not have penicillin hypersensitivity and pseudomonal pneumonia
sqlStmt1 :=
'SELECT ids1.sid patientId
FROM
        ' || src_tab_view || ' ids1, ' || src_tab_view || ' ids2, ' || src_tab_view || ' ids3
         WHERE ids1.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
         AND ids1.oid = ' || to_char(patientClassId,'TM9') || ' AND
         ids2.pid = '|| to_char(rdfTypePropertyId,'TM9') ||'
         AND ids2.oid = ' || to_char(caapClassId,'TM9') || '
         AND ids1.sid = ids3.sid
        AND ids3.pid = ' || to_char(presentPropertyId,'TM9') || '
        AND ids3.oid = ids2.sid

        AND not exists (SELECT 1
        FROM ' || src_tab_view || ' ids4,  ' || src_tab_view || ' ids5
        WHERE ids4.pid = '|| to_char(rdfTypePropertyId,'TM9') || '
        AND ids4.oid = ' || to_char(phClassId,'TM9') || '
        AND ids1.sid = ids5.sid
        AND ids5.pid = ' || to_char(presentPropertyId,'TM9') || '
        AND ids5.oid = ids4.sid )
        AND not exists
        (SELECT 1
        FROM ' || src_tab_view || ' ids6, ' || src_tab_view || ' ids7
        WHERE ids6.pid = '|| to_char(rdfTypePropertyId,'TM9') || '
        AND ids6.oid = ' || to_char(ppClassId,'TM9') || '
        AND ids1.sid = ids7.sid
        AND ids7.pid = ' || to_char(presentPropertyId,'TM9') || '
        AND ids7.oid = ids6.sid )
        ';
        -- insert all eligible patients and recommended regimens
        -- into the model of the ontology
        insertStmt1 := 'INSERT  INTO ' || output_tab || ' (sid, pid, oid)
        SELECT patientId,
        ' || to_char(recomPropertyId,'TM9') || ',
        ' || to_char(medId3, 'TM9') ||'
        FROM (' || sqlStmt1 || ') UNION
        SELECT patientId,
        ' || to_char(recomPropertyId,'TM9') || ',
        ' || to_char(medId4, 'TM9') ||'
        FROM (' || sqlStmt1 || ')
        ';
```

Among the 66 patient medical cases, 64 cases have a very specific antibiotic regimen recommended for patients; 1 case does not have specific antibiotic regimen recommendation but has general medical recommendation for patients (i.e., case 1 in "Hospital Acquired Pneumonia" category); and, 1 case does not have regimen recommendation and general medical recommendation but recommends clinicians to reference another clinical guideline (i.e., case 4 in "Sepsis, Uncertain Focus" category). Therefore, the reasoning rules and functions for finding a regimen recommendation are only defined for those medical cases which have specific regimen recommendations or general medical recommendations for patients. The medical case that has no specific regimen recommendation and general medical recommendation is excluded in the reasoning system. In the QUAIC antibiotic treatment guideline ontology, total 56 rules and 28 inference functions for negation as shown in Table 3 are defined for reasoning of recommended regimens in 65 patient medical cases of Appendix 1.

Disease		**Medical Case No.**	**Rule**	**Inference Function for Negation**
Community Presentation	Sepsis (Uncertain Focus)	1	2	
		2	3	
		3	3	
		4 (No recommendation available)	Nil	Nil
	Febrile Neutropaenia	1		1
		2	1	
		3		1
		4	2	
		5		1
		6	2	
		7		1
		8	2	
	Suspected Fungal Sepsis	1		1
		2		1
		3	1	
		4	1	
	Community Acquired Pneumonia	1		1
		2		1
		3		1
		4		1
		5	2	
		6	1	
	Aspiration	1		1

	Pneumonia	2		1
		3		1
		4		1
		5		1
		6		1
		7		1
		8		1
		9		1
		10	2	
	Suspected Community Acquired Meningitis	1	1	
		2	2	
		3	2	
	Trauma	1	1	
		2	1	
		3	1	
		4	1	
		5	1	
		6	1	
		7	1	
		8	1	
		9	1	
		10	1	
		11	1	
		12	1	
		13	1	
		14	1	
	Urosepsis	1	1	
		2	2	
Healthcare Associated Presentation	Hospital Acquired Pneumonia (HAP)	1 (only general medical recommendations available)	1	
	Early Ventilator Associated Pneumonia (VAP)(provided no known colonisation with MDRO)	1		1
		2	1	
	Late VAP	1		1
		2		1
		3		1
		4		1
		5	1	
	Intra-abdominal	1		1

	Sepsis	2	4	
	Biliary Sepsis (Cholecystitis)	1		1
		2	2	
		3		1
		4	3	
	Acute Pancreatitis	1	1	
Total		**66**	**56**	**28**

Table 3. Rules and Functions for Reasoning in Medical Cases

5.1.2 Query for Finding Recommended Regimen

User defined rules and inferencing functions enable the inference of implicit clinical knowledge in the antibiotic treatment guideline ontology. That is to say if there are some ICU patients have diseases or other clinical presentations described in the patient medical cases, the corresponding rules or functions will fire automatically to infer the recommended regimens provided by that guideline during the reasoning process. This makes it possible to query the inferred recommended regimens for ICU patients. These queries are realised in the Oracle Sem_Match () function which adds SPARQL to SQL to query ontologies.

Queries in Sem_Match () function retrieve the inferred results from the user defined reasoning rules or functions. Some Sem_Match () query examples are used to demonstrate the user defined reasoning rules or reasoning functions for finding recommended regimens and administered antibiotics. Suppose that a rule and an inference function for negation defined in the ontology specify the following two medical cases respectively.

If a patient has febrile neutropenia patient and minor penicillin hypersensitivity, the recommended medication is ceftazidime (200 mg IV 8 hourly)
If a patient has aspiration pneumonia, but has not had penicillin hypersensitivity and pseudomonal pneumonia, the recommended medications are benzylpenicillin (120 mg IV 4 hourly) and metronidazole (500 mg IV 12 hourly);

Assume that some patients (Coy Weston, Irvin Grimer, Margot Potts and Tora Maring) in the ontology have febrile neutropenia and minor penicillin hypersensitivity, and some patients

(Aileen Ashmore, Bell Letchworth and Bettie Flatley) have aspiration pneumonia, but has not had penicillin hypersensitivity and pseudomonal pneumonia. To find what regimen recommendations are available for these patients based on the reasoning rules and functions, the following Sem_Match query could be written to get the inferred results.

```
SELECT patientName, medication, dose_agent, dose_amount, dose_interval,
administration_route, note
FROM TABLE(SEM_MATCH( '{?patient rdfs:label ?patientName.
?patient :has_recommendation ?recommended_medication. ?recommended_medication
rdfs:label ?medication.
?recommended_medication :dose_agent ?dose_agent.

OPTIONAL {?recommended_medication :dose_amount ?dose_amount.}
OPTIONAL {?recommended_medication :interval ?dose_interval. }
OPTIONAL {?recommended_medication :route_of_administration
          ?administration_route.}
OPTIONAL {?recommended_medication :nota_bene ?note. }}',

SEM_MODELS('antibiotics'), SEM_RULEBASES('owl2rl', 'ruelbase1'),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')),
NULL)) ORDER BY patient
```

In this query, a set of triples separated by the period is used to find the patients and the recommended regimens based on the graph pattern matching, the rulebases specified in the query are for reasoning based on the OWL 2 axioms and the user defined reasoning rules and functions. The returned results are shown in Figure 20 and 21.

PATIENTNAME	MEDICATION	DOSE_AGENT	DOSE_AMOUNT	DOSE_INTERVAL	ADMINISTRATION_ROUTE	NOTE
Coy Weston	Medication 2 (Febrile Neutropaenia)	Ceftazidime	200 mg	8 hours	Intravenous	(null)
Irvin Grimmer	Medication 2 (Febrile Neutropaenia)	Ceftazidime	200 mg	8 hours	Intravenous	(null)
Margot Potts	Medication 2 (Febrile Neutropaenia)	Ceftazidime	200 mg	8 hours	Intravenous	(null)
Tora Maring	Medication 2 (Febrile Neutropaenia)	Ceftazidime	200 mg	8 hours	Intravenous	(null)

Figure 20. Regimen Recommendation Query Result for Febrile Neutropaenia Patients

PATIENTNAME	MEDICATION	DOSE_AGENT	DOSE_AMOUNT	DOSE_INTERVAL	ADMINISTRATION_ROUTE	NOTE
Aileen Ashmore	Medication 2 (Aspiration Pneumonia)	Metronidazole	500 mg	12 hours	Intravenous	(null)
Aileen Ashmore	Medication 1 (Aspiration Pneumonia)	Benzylpenicillin	120 mg	4 hours	Intravenous	(null)
Bell Letchworth	Medication 1 (Aspiration Pneumonia)	Benzylpenicillin	120 mg	4 hours	Intravenous	(null)
Bell Letchworth	Medication 2 (Aspiration Pneumonia)	Metronidazole	500 mg	12 hours	Intravenous	(null)
Hettie Flatley	Medication 2 (Aspiration Pneumonia)	Metronidazole	500 mg	12 hours	Intravenous	(null)
Hettie Flatley	Medication 1 (Aspiration Pneumonia)	Benzylpenicillin	120 mg	4 hours	Intravenous	(null)
Kevin Majewski	Medication 2 (Aspiration Pneumonia)	Metronidazole	500 mg	12 hours	Intravenous	(null)
Kevin Majewski	Medication 1 (Aspiration Pneumonia)	Benzylpenicillin	120 mg	4 hours	Intravenous	(null)

Figure 21. Regimen Recommendation Query Result for Aspiration Pneumonia Patients

5.2 Temporal Knowledge Related Reasoning in the Antibiotic Treatment Guideline Ontology

Many traditional OWL-based clinical guideline ontology systems focus on the clinical knowledge reasoning to find various guideline recommendations for clinicians, but these systems often ignore the temporal knowledge reasoning due to the limitation of OWL. The extended 4D fluent representation method presented in this thesis enables the temporal knowledge representation and reasoning in the antibiotic treatment guideline ontology such that it can overcome the shortcoming of traditional OWL-based guideline systems to an extent. This section analyses how this extended 4D fluent modelling approach realises the temporal knowledge reasoning in the antibiotic treatment guideline ontology.

5.2.1 Query for Finding Administered Antibiotics

In addition to finding the recommended regimens in the guideline, clinicians are also likely to find the actually administered antibiotics for ICU patients and compare them with these recommendations to check if any different drugs are administered to the patients. Since ICU patients are usually administered antibiotics in different time periods, under the 4D fluent temporal knowledge modelling method as analysed in previous chapters, the “administered with” relation between a patient and an antibiotic becomes the relation between the temporal parts or time slices of these two participating entities. Therefore, query of this type of clinical knowledge involves temporal knowledge related graph pattern matching which is based on the extended 4D fluent method.

The following query is for finding the administered antibiotics which are same as the recommended ceftazidime antibiotic in the regimen recommendation for the patients. The graph pattern matching in the first filter part of the query is about the “administered with” relation between the patients and the administered antibiotics as analysed above. Temporal parts of patients and administered antibiotics are connected via the “administered with” relation in the filter clause.

```
SELECT patient_label, administered_drug$_suffix, dose_amount, admin_route
FROM TABLE(SEM_MATCH('{ ?patient rdfs:label ?patient_label.

OPTIONAL {?administered_drug :dose_amount ?dose_amount.}
OPTIONAL {?administered_drug :route_of_administration ?admin_route. }
FILTER (EXISTS {?patient :has_temporal_part ?patient_tp.
?administered_drug :has_temporal_part ?drug_tp.
?patient_tp :administered_with ?drug_tp.})
FILTER (EXISTS {?patient :has_recommendation ?recommended_medication. })
FILTER (EXISTS {?administered_drug rdf:type :Ceftazidime}) }',

SEM_MODELS('antibiotics'),
SEM_RULEBASES('owl2rl', 'rulebase1'),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')),
 NULL)) ORDER BY patient, administered_drug
```

PATIENT_LABEL	ADMINISTERED_DRUG$_SUFFIX	DOSE_AMOUNT	ADMIN_ROUTE
Coy Weston	ceftazidime13	100 mg	oral
Coy Weston	ceftazidime2	200 mg	intravenous
Irvin Grimmer	ceftazidime1	200 mg	intravenous
Irvin Grimmer	ceftazidime11	200 mg	intravenous
Irvin Grimmer	ceftazidime12	150 mg	intravenous
Margot Potts	ceftazidime14	200 mg	intravenous
Margot Potts	ceftazidime15	150 mg	intravenous
Margot Potts	ceftazidime8	200 mg	intravenous
Margot Potts	ceftazidime9	200 mg	intravenous
Tora Maring	ceftazidime16	200 mg	intravenous
Tora Maring	ceftazidime3	150 mg	oral
Tora Maring	ceftazidime4	180 mg	oral
Tora Maring	ceftazidime5	150 mg	oral

Figure 22. Administered Antibiotics which are in the Recommended Regimen

Similarly, this query below is for finding the administered antibiotics which are not the same as the ceftazidime antibiotic in the regimen recommendation for the patients.

```
SELECT patient_label, administered_drug$_suffix, dose_amount, admin_route
FROM TABLE(SEM_MATCH('{ ?patient rdfs:label ?patient_label.

OPTIONAL {?administered_drug :dose_amount ?dose_amount.}
OPTIONAL {?administered_drug :route_of_administration ?admin_route. }
FILTER (EXISTS {?patient :has_temporal_part ?patient_tp.
?administered_drug :has_temporal_part ?drug_tp.
?patient_tp :administered_with ?drug_tp.})
FILTER (EXISTS {?patient :has_recommendation ?recommended_medication. })
FILTER (NOT EXISTS {?administered_drug rdf:type :Ceftazidime} ) }',

SEM_MODELS('antibiotics'),
SEM_RULEBASES('owl2rl', 'rulebase1'),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')),
NULL)) ORDER BY patient, administered_drug
```

PATIENT_LABEL	ADMINISTERED_DRUG$_SUFFIX	DOSE_AMOUNT	ADMIN_ROUTE
Coy Weston	neomycin2	200 mg	intravenous
Coy Weston	vancomycin34	150 mg	intravenous
Coy Weston	vancomycin8	150 mg	intravenous
Irvin Grimmer	paromomycin2	50 mg	oral
Irvin Grimmer	vancomycin25	200 mg	intravenous
Irvin Grimmer	vancomycin7	150 mg	intravenous
Margot Potts	ceftibuten2	150 mg	oral
Margot Potts	ceftibuten3	100 mg	oral
Tora Maring	vancomycin10	120 mg	oral
Tora Maring	vancomycin11	100 mg	oral
Tora Maring	vancomycin38	150 mg	intravenous
Tora Maring	vancomycin9	150 mg	intravenous

Figure 23. Administered Antibiotics which are not in the Recommended Regimen

5.2.2 Compliance Checking of Dose Interval and Dose Duration

Periodical dose interval is a very important repetitive temporal constraint in the guideline regimen recommendations. In this antibiotic treatment guideline, each recommended antibiotic has a periodical dose interval temporal constraint for the recommended antibiotics whereas some of them have both periodical dose interval and dose duration temporal constraints. In real clinical practice, they are particularly important for guideline compliance checking for drug administration where fixed periodical intervals between doses need to be followed for safety and efficacy purposes. In order to calculate the dose interval and dose duration of administered antibiotics, two inference functions are defined respectively (Appendix 10 and 11) for related reasoning.

The main part of the function for calculating dose interval consists of two SQL select queries as shown below. The first SQL query is used to extract the ID of each dose interval and its open endpoint and close endpoint, and the time value of each endpoint. The second query is used to calculate the difference in hours between the time value of open endpoint and the time value of close endpoint for that interval. Similarly, in the main part of the function for calculating dose duration, the first query is used to extract the ID of each dose period and its start time point and end time point, and the time value of its start time and end time. The second query is used to calculate the difference in days between its start time and end time.

```
-- extract the ID of each dose interval and its open and close endpoints,
-- and the time value of each endpoint.
sqlStmt := ‘SELECT ids1.sid timeIntervalInstance, ids2.sid openTimeInstant, ids3.sid
closeTimeInstant,
TO_TIMESTAMP_TZ(values1.value_name,''YYYY-MM-DD"T"HH24:MI:SSTZH:TZM'')
openTime,
TO_TIMESTAMP_TZ(values2.value_name,''YYYY-MM-DD"T"HH24:MI:SSTZH:TZM'')
closeTime
FROM ' || resource_id_map_view || ' values1,
' || resource_id_map_view || ' values2,
' || src_tab_view || ' ids1, ' || src_tab_view || ' ids2,
' || src_tab_view || ' ids3, ' || src_tab_view || ' ids4,
' || src_tab_view || ' ids5, ' || src_tab_view || ' ids6,
' || src_tab_view || ' ids7

WHERE ids1.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND ids1.oid = ' || to_char(timeIntervalClassId,'TM9') || '
AND ids2.pid = '|| to_char(rdfTypePropertyId,'TM9')||'
AND ids2.oid = '|| to_char(timeInstantClassId,'TM9') ||'
AND ids3.pid ='|| to_char(rdfTypePropertyId,'TM9')||'
AND ids3.oid = '|| to_char(timeInstantClassId,'TM9') ||'
AND ids4.sid = ids1.sid
AND ids4.pid = ' || to_char(openInstantPropertyId,'TM9') || '
AND ids4.oid = ids2.sid
AND ids5.sid = ids1.sid
AND ids5.pid = ' || to_char(closeInstantPropertyId,'TM9') || '
AND ids5.oid = ids3.sid
AND ids6.sid = ids2.sid
AND ids6.pid = ' || to_char (dateTimeValPropertyId,'TM9') || '
AND ids6.oid = values1.value_id
AND ids7.sid = ids3.sid
AND ids7.pid = ' || to_char (dateTimeValPropertyId,'TM9') || '
AND ids7.oid = values2.value_id

-- compute the difference (in hours) between the two timestamps
-- from the sqlStmt query. store the hours as xsd:decimal.
insertStmt := 'INSERT  INTO ' || output_tab || ' (sid, pid, o)
SELECT timeIntervalInstance, ' || to_char(timeIntervalPropertyId,'TM9') || ',
          '''' || hours || ''''^^xsd:decimal''
FROM (SELECT timeIntervalInstance, ( trunc(
    (extract(day from (closeTime - openTime))*24 +
     extract (hour from (closeTime - openTime)) +
     extract (minute from (closeTime - openTime))/60),1) ) hours
     FROM (' || sqlStmt || '))';
```

By leveraging the inference functions for calculating dose interval and dose duration, and the extended 4D fluent approach, all dose intervals and dose durations of administered antibiotics for ICU patients can be retrieved and compared with the recommended dose interval and dose duration temporal constraints in the regimens for compliance checking. The following two Sem_Match () queries are the examples for retrieving the calculated dose intervals and those which are not the same as the recommended interval. For the dose duration, the queries are very similar to the ones for dose interval and are omitted here. Graph pattern matching in the queries involves the temporal parts of participating entities in the "administered_with" relation, dose interval, the open and close endpoints of dose interval, and the periodical dose interval temporal constraint in the recommendation. In the previous example of patients who have febrile neutropenia patient and minor penicillin hypersensitivity in section 5.1.2, the query and returned results for the dose interval of administered antibiotics are shown below.

```
SELECT patient_label, administered_drug$_suffix, interval_start, interval_end,
interval_length_hours
FROM TABLE(SEM_MATCH('{ ?patient rdfs:label ?patient_label.
?patient_tp_open :has_temporal_entity ?open_instant.
?patient_tp_close :has_temporal_entity ?close_instant.
?patient :has_temporal_part ?patient_tp_open.
?patient :has_temporal_part ?patient_tp_close.
?administered_drug :has_temporal_part ?drug_tp_open.
?drug_tp_open :has_temporal_entity ?open_instant.
?administered_drug :has_temporal_part ?drug_tp_close.
?drug_tp_close :has_temporal_entity ?close_instant.
?patient_tp_open :administered_with ?drug_tp_open.
?patient_tp_close :administered_with ?drug_tp_close.
?dose_interval :interval_hourly ?interval_length_hours.
?dose_interval :open_instant ?open_instant.
?dose_interval :close_instant ?close_instant.
?open_instant :dateTimeValue ?interval_start.
?close_instant :dateTimeValue ?interval_end.

FILTER (exists {?patient :has_recommendation ?recommended_medication. } )}',

SEM_MODELS('antibiotics'), SEM_RULEBASES('owl2rl', 'rulebase1'),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')),
NULL)) ORDER BY patient, administered_drug, dose_interval
```

PATIENT_LABEL	ADMINISTERED_DRUG$_SUFFIX	INTERVAL_START	INTERVAL_END	INTERVAL_LENGTH_HOURS
Irvin Grimmer	ceftazidime1	2010-03-07T14:00:00	2010-03-07T23:30:00	9.5
Margot Potts	ceftazidime9	2010-03-18T09:00:00	2010-03-18T17:25:00	8.4
Margot Potts	ceftazidime9	2010-03-18T17:25:00	2010-03-19T01:55:00	8.5
Tora Maring	ceftazidime3	2010-04-11T09:40:00	2010-04-11T17:35:00	7.9
Tora Maring	ceftazidime3	2010-04-10T08:00:00	2010-04-10T16:25:00	8.4
Tora Maring	ceftazidime3	2010-04-10T16:25:00	2010-04-11T01:30:00	9
Tora Maring	ceftazidime3	2010-04-11T01:30:00	2010-04-11T09:40:00	8.1
Tora Maring	ceftazidime3	2010-04-11T17:35:00	2010-04-12T01:45:00	8.1
Tora Maring	ceftazidime4	2010-04-20T14:00:00	2010-04-20T22:10:00	8.1

Figure 24. Dose Intervals of Administered Antibiotics

It is also possible to check the dose intervals of administered antibiotics which do not meet the dose interval temporal constraint requirement in the guideline. The query below compares the dose intervals of administered ceftazidime antibiotic with the 8 hours dose interval requirement recommended by the guideline to find any dose intervals which are not the same as the recommended one. The returned result is shown in Figure 25.

```
SELECT patient_label, administered_drug$_suffix, interval_start, interval_end,
interval_length_hours
FROM TABLE(SEM_MATCH('{ ?patient rdfs:label ?patient_label.
?patient_tp_open :has_temporal_entity ?open_instant.
?patient_tp_close :has_temporal_entity ?close_instant.
?patient :has_temporal_part ?patient_tp_open.
?patient :has_temporal_part ?patient_tp_close.
?administered_drug :has_temporal_part ?drug_tp_open.
?drug_tp_open :has_temporal_entity ?open_instant.
?administered_drug :has_temporal_part ?drug_tp_close.
?drug_tp_close :has_temporal_entity ?close_instant.
?patient_tp_open :administered_with ?drug_tp_open.
?patient_tp_close :administered_with ?drug_tp_close.
?dose_interval :open_instant ?open_instant.
?dose_interval :close_instant ?close_instant.
?dose_interval :interval_hourly ?interval_length_hours.
?open_instant :dateTimeValue ?interval_start.
?close_instant :dateTimeValue ?interval_end.
?pi rdf:type :Periodical_Interval.
?pi :periodical_interval_constraint_length ?piv.

FILTER (exists {?patient :has_recommendation ?recommended_medication. } )
FILTER (exists {?administered_drug rdf:type :Ceftazidime. } )
FILTER (sameTerm (?pi, :every8hrs)) FILTER (?interval_length_hours != ?piv) }',

SEM_MODELS('antibiotics'),
SEM_RULEBASES('owl2rl', 'rulebase1'),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')),
NULL)) ORDER BY patient, administered_drug
```

PATIENT_LABEL	ADMINISTERED_DRUG$_SUFFIX	INTERVAL_START	INTERVAL_END	INTERVAL_LENGTH_HOURS
Irvin Grimmer	ceftazidime1	2010-03-07T14:00:00	2010-03-07T23:30:00	9.5
Margot Potts	ceftazidime9	2010-03-18T09:00:00	2010-03-18T17:25:00	8.4
Margot Potts	ceftazidime9	2010-03-18T17:25:00	2010-03-19T01:55:00	8.5
Tora Maring	ceftazidime3	2010-04-11T09:40:00	2010-04-11T17:35:00	7.9
Tora Maring	ceftazidime3	2010-04-10T08:00:00	2010-04-10T16:25:00	8.4
Tora Maring	ceftazidime3	2010-04-10T16:25:00	2010-04-11T01:30:00	9
Tora Maring	ceftazidime3	2010-04-11T01:30:00	2010-04-11T09:40:00	8.1
Tora Maring	ceftazidime3	2010-04-11T17:35:00	2010-04-12T01:45:00	8.1
Tora Maring	ceftazidime4	2010-04-20T14:00:00	2010-04-20T22:10:00	8.1

Figure 25. Dose Intervals of Administered Ceftazidime which are different than the 8 Hours Dose Interval Requirement in the Guideline

5.2.3 Temporal Relation Reasoning in Administered Antibiotics

In addition to dose interval and dose duration, temporal relation is also an important factor in antibiotic administration. The ordering or sequence of antibiotic administration can have different clinical implications. According to a study on the simultaneous and staggered administration on combination regimens against *Pseudomonas aeruginosa* in an in vitro infection mode conducted by Zelenitsky et al. in [49], antibiotic sequence has impact on the effect of administered antibiotics. Their study shows that simultaneous dosing of ceftazidime and ciprofloxacin or ceftazidime and tobramycin at 24, 36 and 48 hours interval is significantly more active in bacterial kill than the dosing of ceftazidime followed by the dosing of ciprofloxacin or tobramycin at a 1.5 hours interval. Their study also shows that antibiotic sequence has a significant and class-dependent effect on antibacterial response. Ceftazidime combined with ciprofloxacin or tobramycin was more active if ceftazidime was administered before or with the other antibiotics. Therefore, finding the temporal relations between administered antibiotics in this guideline ontology can help clinicians make decisions.

Previous analysis has showed that the extended 4D fluent ontology enables the computation of dose interval and dose duration of administered antibiotics and the compliance checking in terms of these temporal constraints specified in that antibiotic treatment guideline. However, in order to implement the reasoning over the temporal relations between administered antibiotics, it is necessary to define these relations and related reasoning rules in the ontology. These temporal relations and reasoning rules are based on Allen's interval algebra.

5.2.3.1 Temporal Relations in Allen's Interval Algebra

A time interval is an ordered pair of $<X^-, X^+>$ where X^- (start time or open endpoint) $< X^+$ (end time or close endpoint). Allen in [50] defined 13 mutually exclusive basic temporal relations for time intervals. Among these relations, 6 pairs are the inverses which are <before, after>, <meet, met by>, <overlap, overlapped by>, <during, contain>, <start, started by> and <finish, finished by>. Moreover, each basic relation can be defined in terms of the relations between the endpoints of intervals. These 13 basic relations are summarised in Table 4 which is modified from the table in [50] and the table in [51].

Basic Relations	Symbol	Illustration	Endpoint Relations
X before Y Y after X	b bi	XXXX YYYY	$X^+ < Y^-$
X meet Y Y met by X	m mi	XXXX YYYY	$X^+ = Y^-$
X overlap Y Y overlapped by X	o oi	XXXX YYYY	$X^- < Y^-$, $X^+ > Y^-$, $X^+ < Y^+$
X during Y Y contain X	d di	XXXX YYYYYYYY	$X^- > Y^-$, $X^+ < Y^+$
X start Y Y started by X	s si	XXXX YYYYYYYY	$X^- = Y^-$, $X^+ < Y^+$
X finish Y Y finished by X	f fi	XXXX YYYYYYYY	$X^- > Y^-$, $X^+ = Y^+$
X equal Y	eq	XXXX YYYY	$X^- = Y^-$, $X^+ = Y^+$

Table 4. Allen's 13 Basic Relations

The 13 basic relations describe the definite temporal relations between intervals. However, there are also indefinite fuzzy temporal relations between intervals. An indefinite temporal relation is a disjunction of basic relations which can be represented as a relation set. For example, the relation "before or meet or overlap" is the disjunction of before, meet and overlap, and is represented as {b, m, o}. The order of these relations in the set is irrelevant. 2^{13} (8192) temporal relation sets can be yielded based on all possible disjunctions of these 13 relations. Except the relation sets which contain only one basic relation and an empty relation set $\varnothing$, the rest of them are the indefinite temporal relations [51]. Among these indefinite temporal relations, the disjunction of all 13 basic relations is also called "full" relation whereas the disjunction of relations of "overlap", "finished-by", "contain", "start", "equal", "started-by", "during", "finish" and "overlapped-by" is also called "concur" relation [52].

There are five common operations on these basic relations which are union, intersection, composition, inverse (converse) and complement [52] which are related to the temporal relation reasoning. For sake of convenience, the abbreviations of these relation names are used in these operations. The explanation of these operations is shown below.

- Union of two relation sets (∪)
    - It yields a collection of all relation components in that two sets.
    - E.g., {b, m, o} ∪ {d, s, f} −> {b, m, o, d, s, f}.
- Intersection of two relation sets (∩)
    - It yields a collection of the common relation components in that two sets.
    - E.g., {b, m, o} ∩ {d, o, s} is {o}.
- Composition of two relation sets (○)
    - It yields a new relation.
    - E.g., assume that there are three time intervals X, Y and Z in which X is before ({b}) Y and Y meets ({m}) Z. Then, {b} ○ {m} -> {b}, i.e., X is before Z.
- Inverse operation on a relation (!)
    - It reverses the relation.
    - E.g., !{b} -> {bi}.
- Complement of a relation set (~)
    - It yields a collection of all relation components which are not in that relation set.
    - E.g., ~{b, m, o} -> {bi, mi, oi, d, di, s, si, f, fi, eq}.

Allen in [50] proposes a transitivity table which lists the relations yielded from the composition of any two basic relations with omitting the “equal” relation. The following compositions in Table 5 are based on the composition table in [52] which includes the “equal” relation.

	b	**m**	**o**	**fi**	**di**	**s**	**eq**	**si**	**d**	**f**	**oi**	**mi**	**bi**
b	b	b	b	b	b	b	b	bi	b,m,o, s,d	b,m,o, s,d	b,m,o, s,d	b,m,o, s,d	full
m	b	b	b	b	b	m	m	m	o,s,d	o,s,d	o,s,d	fi,eq,f	di,si,oi, mi,bi
o	b	b	b,m,o	b,m,o	b,m,o, fi,di	o	o	o,fi,di	o,s,d	o,s,d	concur	di,si,oi	di,si,oi, mi,bi
fi	b	m	o	fi	di	o	fi	di	o,s,d	fi,eq,f	di,si,oi	di,si,oi	di,si,oi, mi,bi
di	b,m,o, fi,di	o,fi,di	o,fi,di	di	di	o,fi,di	di	di	concur	di,si,oi	di,si,oi	di,si,oi	di,si,oi, mi,bi
s	b	b	b,m,o	b,m,o	b,m,o, fi,di	s	s	s,eq,si	d	d	d,f,oi	mi	bi
eq	b	m	o	fi	di	s	eq	si	d	f	oi	mi	bi
si	b,m,o, fi,di	o,fi,di	o,fi,di	di	di	s,eq,si	si	si	d,f,oi	oi	oi	mi	bi
d	b	b	b,m,o, s,d	b,m,o, s,d	full	d	d	d,f,oi, mi,bi	d	d	d,f,oi, mi,bi	bi	bi
f	b	m	o,s,d	fi,eq,f	di,si,oi, mi,bi	d	f	oi,mi,bi	d	f	oi,mi,bi	bi	bi
oi	b,m,o, fi,di	o,fi,di	concur	di,si,oi	di,si,oi, mi,bi	d,f,oi	oi	oi,mi,bi	d,f,oi	oi	oi,mi,bi	bi	bi
mi	b,m,o, fi,di	s,eq,si	d,f,oi	mi	b	d,f,oi	mi	bi	d,f,oi	mi	b	bi	bi
bi	full	d,f,oi, mi,bi	d,f,oi, mi,bi	bi	b	d,f,oi, mi,bi	bi	bi	d,f,oi, mi,bi	bi	b	bi	bi

Table 5. Composition of Two Basic Relations for Three Intervals

In this table, there are 169 (13 × 13) compositions of two basic relations for three time intervals which yields 27 unique relations in which 13 of them are the basic relations and 14 are the indefinite fuzzy relations. Among these 14 indefinite relations, 5 pairs of relations are the inverses. The indefinite relations except the "full" relation can also be defined in terms of the endpoint relations of intervals which are shown in Table 6.

Fuzzy Relations	**Endpoint Relations**
X {d, f, oi} Y Y {di, fi, oi} X	$X^- > Y^-, X^- < Y^+$
X {b, m, o, fi, di} Y Y {bi, mi, oi, f, d} X	$X^- < Y^-$
X {b, m, o} Y Y {bi, mi, oi} X	$X^- < Y^-, X^+ < Y^+$
X {b, m, o, s, d} Y Y {bi, mi, oi, si, di} X	$X^+ < Y^+$
X {d, o, s} Y Y {di, oi, si} X	$X^+ > Y^-, X^+ < Y^+$
X {si, eq, s} Y	$X^- = Y^-$
X {f, eq, fi} Y	$X^+ = Y^+$
X concur Y	$X^- < Y^+, X^+ > Y^-$

Table 6. The Indefinite Relations from the Composition of Two Basic Relations

Each composition in Table 5 implies a rule which can be used for temporal relation reasoning in the ontology. For example, the composition {b} ○ {m} yields a definite relation {b} and the composition {b} ○ {d} yields an indefinite fuzzy relation {b, m, o, s, d}. For the former composition, a rule could be defined to state if interval X is before Y and Y meets Z, then X is before Z. For the latter one, it could be a rule which states if interval X is before Y and Y is during Z, then X is before, meets, overlaps, starts or is during Z.

Since there are 8192 relations in the full algebra, all possible compositions of relation pairs except the empty relation in this set are $(2^{13}-1) \times (2^{13}-1)$, i.e., 67,092,481 [53]. However, determining the satisfiability of an arbitrary collection of relations on intervals in the full algebra is NP-complete (i.e., intractable, no polynomial time algorithm or fast and efficient solution exists). Krokhin, Jeavons and Jonsson in [54] show that there are 18 maximal tractable subsets of the full algebra where the polynomial time algorithm exists for

temporal reasoning. Reasoning in any fragment of the full algebra which is not entirely contained in one of these maximal tractable subsets is NP-complete. Among the 18 subsets, the ORD Horn subalgebra is the smallest one which has 868 elements, but it is also the only one containing all 13 basic relations. However, the number of compositions of all relation pairs in this set is 753,424 (868 × 868). The continuous endpoint subclass and the pointisable subclass of ORD Horn subalgebra contain 83 and 188 relations respectively [51]. The number of compositions of all relation pairs in these two sets is 6,889 (83 × 83) and 35,344 (188 × 188). However, implementation of any tractable subset analysed above in an ontology will need very large amount of reasoning rules to be written for that ontology. Reasoning efficiency will also decrease as large amount rules and time intervals involved in the ontology. Considering these issues, temporal reasoning in the antibiotic treatment guideline ontology is restricted to the compositions listed in Table 5.

5.2.3.2 Constraint Propagation Algorithm in Allen's Interval Algebra

Allen presents a constraint propagation algorithm in [50] to compute the transitive closure of constraints about temporal relations on intervals. It is used widely for temporal reasoning in various domains. This algorithm repetitively applies the composition and intersection operations on temporal relations between intervals. The composition operation is realised in a constraint function which performs the composition and union operation on each element in two relation sets for three intervals to get the inferred relation. Then, the algorithm repetitively applies the intersection operation and the constraint function to compute the transitive closure of constraints. The core part of this algorithm is summarised in the following formula [55] in which R_{ik} denotes the relation between i and k, R_{ij} denotes the relation between i and j, and R_{jk} denotes the relation between j and k.

$$\forall i, \forall j, \forall k \; (R_{ik} \leftarrow R_{ik} \cap R_{ij} \circ R_{jk})$$

For example, assume that there are three relations R_1 ({d, o, s}) between interval i and j, R_2 ({d}) between j and k, and R_3 ({eq, d, di, o, oi, s}) between i and k. The steps [56] of applying these operations to find the relation between and i and k are shown below.

- {eq, d, di, o, oi, s} ∩ {d, o, s} ∘ {d}
- {eq, d, di, o, oi, s} ∩ ({d} ∘ {d} ∪ {o} ∘ {d} ∪ {s} ∘ {d})
- {eq, d, di, o, oi, s} ∩ ({d} ∪ {d, o, s} ∪ {d})
- {eq, d, di, o, oi, s} ∩ {d, o, s}
- {d, o, s}

This algorithm firstly applies the composition and union operation on the relations R_1 and R_2 in the right side to get the inferred relation between i and k. If there is already a relation R_3 between i and k, then it applies the intersection operation between the relation R_3 and the inferred relation from the composition operation to check if a conflict exists. If the intersection yields a result set containing one element, then the constraint between i and k has been uniquely determined, i.e., the relation between i and k is a definite basic relation; if the result set contains more than one relations, the relation is an indefinite fuzzy relation; if the result set is empty, then a conflict exists in the temporal network. As to the example mentioned above, the inferred exact relation between i and k is {d, o, s}.

As can be seen in this algorithm, temporal reasoning basically includes two tasks: finding the implicit temporal relation in the network and finding the inconsistencies in the network. This algorithm is implemented in a rule-based approach in the antibiotic treatment guideline ontology to find the exact temporal relations between administered drugs and check the potential inconsistent temporal relations in the antibiotic treatment guideline ontology.

5.2.3.3 Finding the Temporal Relations between Administered Antibiotics

In table 5, there are 27 unique relations yielded from the composition of basic relations in which 13 of them are the basic relations and 14 are the indefinite fuzzy relations. These 27 relations are organised under two top level object properties namely “allenBasic” and “allenFuzzy” in the ontology (Appendix 5). These 27 relations except the “full” relation can also be defined by the endpoint relations of intervals as shown in Table 4 and Table 6. A set of Oracle user defined rules are used to define these basic relations and fuzzy relations based on endpoint relations.

Among the 13 basic relations, 6 pairs are the inverses as shown in Table 3. By leveraging the inverse object property axiom of OWL, only 7 of 13 basic relations are needed to be defined using Oracle user defined rules in terms of endpoint relations. These 7 basic relations are "before", "meet", "overlap", "during", "start", "finish" and "equal". For example, the "before" relation can be defined in the following user defined rule in terms of endpoint relations. The definitions of all basic relations are listed in Appendix 12.

```
INSERT INTO mdsys.semr_allenBasic VALUES(
'defBefore',
'(?t1 rdf:type :Time_Period) (?t1 :close_instant ?closeInstant1) (?closeInstant1
:dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2
:dateTimeValue ?ov2) ',
'(cv1 < ov2)',
'(?t1 :before ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

Similarly, among the 14 indefinite fuzzy relations, 5 pairs are the inverses as shown in Table 5. By leveraging the inverse object property axiom of OWL, only 8 of them are needed to be defined in terms of endpoint relations using Oracle user defined rules. These 8 fuzzy relations are {d, f, oi}, {b, m, o, fi, di}, {b, m, o}, {b, m, o, s, d}, {d, o, s}, {si, eq, s}, {f, eq, fi} and concur. For example, the {d, f, oi} relation, which represents the "during or finish or overlapped by" relation, can be defined in the following user defined rule in terms of endpoint relations. The definitions of all indefinite fuzzy relations are listed in Appendix 12.

```
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_d_f_oi',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?openInstant1
:dateTimeValue ?ov1) (?t1 :close_instant ?closeInstant1) (?closeInstant1
:dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2
:dateTimeValue ?ov2) (?t2 :close_instant ?closeInstant2) (?closeInstant2
:dateTimeValue ?cv2) ',
'((ov1 > ov2) and (ov1 < cv2))',
'(?t1 :during_or_finish_or_overlapped_by ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

In order to find all exact temporal relations between administered antibiotics, it is necessary to define the temporal reasoning rules for Allen's relations. Although OWL 2 has a set of axioms about object properties which can provide the support for temporal reasoning in

Allen's relations, the global restrictions on these axioms in OWL 2 also limit the reasoning capability of OWL in order to ensure the decidability of OWL reasoning [13]. In OWL 2, an object property is composite if it meets one of the following conditions:

- It is equal to owl:topObjectProperty or owl:bottomObjectProperty.
- It is a transitive object property or transitive of the inverse of an object property.
- It can be inferred from composition of two or more other object properties by means of a property chain.

An object property expression is simple if it has no direct or indirect subproperties that are either transitive or are defined by means of property chains. However, OWL 2 does not allow composite properties to be functional, inverse functional, irreflexive, and asymmetric. OWL 2 also does not allow property disjointness, cardinality restrictions and self-restriction on composite properties.

In Allen's interval algebra, all of the 13 basic relations are mutually exclusive, i.e., each of them is disjoint of another. Moreover, the relations including "before", "after", "contain", "during", "finish", "finished by", "start" and "started by" are transitive. The relation "equal" is transitive and symmetric. However, OWL 2 specification does not allow a transitive property to be disjoint of another property since it is a composite relation. For example, it violates the global restriction of OWL 2 if the transitive relation "before" is defined to be disjoint of the "meet" relation in the ontology. OWL 2 specifications also do not allow a property to be disjoint of another property if it is inferred from the composition of two or more other properties by means of a property chain in the ontology. For example, it violates the global restriction of OWL 2 if the relation "before" is defined to be disjoint of the relation "overlap" and it is also inferred from the composition of "before" and "meet" by means of a property chain. Therefore, it is not feasible to implement all temporal reasoning features for Allen's relations within OWL itself.

However, these global restrictions do not apply to rule languages such as SWRL and Oracle user defined rules. In order to avoid the violation of global restriction of OWL 2 in the ontology, Oracle user defined rules are used to implement the temporal reasoning in Allen's relations in the ontology. These reasoning rules are based on the compositions in Table 5. As

can be seen in Table 5, there are 169 compositions in the table in which 97 compositions yield the basic relations and 72 compositions yields the indefinite fuzzy relations. Theoretically, it can have 169 rules to be defined for reasoning; but, it is not necessary. By leveraging some important OWL reasoning features such as axioms on inverse, transitive and symmetric object properties, a significant number of rules can be reduced to improve the reasoning performance. Among the 97 compositions for the basic relations, the relations “after”, “contain”, “overlapped by”, “started by” and “finished by” can be defined as the inverses of “before”, “during”, “overlap”, “start”, and “finish” respectively. Moreover, “before”, “after”, “contain”, “during”, “finish”, “finished by”, “start” and “started by” can be defined as the transitive object properties. In addition, the “equal” relation can be defined as a transitive and symmetric property. As a result, only 44 rules are defined for the relations “before”, “during”, “overlap”, “start” and “finish” in the ontology. Among the 72 compositions for the 14 fuzzy relations, {b, m, o}, {b, m, o, s, d}, {b, m, o, fi, di}, {d, f, oi} and {d, o, s} can be defined as the inverses of {bi, mi, oi}, {bi, mi, oi, si, di}, {bi, mi, oi, fi, di}, {di, fi, o} and {di, oi, si} respectively. As result, only 42 rules are defined for {b, m, o}, {b, m, o, fi, di}, {b, m, o, s, d}, concur, {d, f, oi}, {d, o, s}, {f, eq, fi}, full and {si, eq, s} in the ontology. Totally, 86 rules are defined for temporal relation reasoning. The compositions used for the 86 rules are listed in Table 7.

Composition (86)	**Inferred Relation**
{m} ○ {m}; {o} ○ {b}; {eq} ○ {b}; {d} ○ {b}; {b} ○ {o}; {s} ○ {b}; {b} ○ {fi}; {fi} o {b}; {o} ○ {m}; {d} ○ {m}; {b} ○ {eq}; {b} o {m}; {b} ○ {s}; {m} ○ {di}; {b} ○ {di}; {f} ○ {b}; {b} ○ {si}; {m} ○ {fi}; {s} ○ {m}; {m} ○ {o}; {m} ○ {b}	{b}
{d} ○ {s}; {f} ○ {s}; {s} ○ {f}; {f} ○ {d}; {d} ○ {f}; {eq} ○ {d}; {s} ○ {d}; {d} ○ {eq}	{d}
{f} ○ {m}; {m} ○ {s}; {fi} ○ {m}; {m} ○ {si}; {m} ○ {eq}; {eq} ○ {m}	{m}
{fi} ○ {s}; {o} ○ {eq}; {o} ○ {s}; {fi} ○ {o}; {eq} ○ {o}	{o}
{s} ○ {eq}; {eq} ○ {s}	{s}
{eq} ○ {f}; {f} ○ {eq}	{f}
{o} ○ {fi}; {s} ○ {fi}; {s} ○ {o}	{b, m, o}
{di} ○ {b}; {oi} ○ {b}; {mi} ○ {b}; {o} ○ {di}; {s} ○ {di}; {si} ○ {b}	{b, m, o, fi, di}
{b} ○ {d}; {b} ○ {f}; {d} ○ {fi}; {d} ○ {o}; {b} ○ {mi}; {b} ○ {oi}	{b, m, o, s, d}
{o} ○ {oi}; {di} ○ {d}; {oi} ○ {o}	concur
{si} ○ {d}; {oi} ○ {s}; {oi} ○ {d}; {mi} ○ {s}; {mi} ○ {o}; {s} ○ {oi}; {mi} ○ {d}	{d, f, oi}
{m} ○ {d}; {f} ○ {o}; {o} ○ {f}; {m} ○ {oi}; {fi} ○ {d}; {m} ○ {f}; {o} ○ {d}	{d, o, s}
{f} ○ {fi}; {m} ○ {mi}; {fi} ○ {f}	{f, eq, fi}
{b} ○ {bi}; {bi} ○ {b}; {d} ○ {di}	full
{mi} ○ {m}; {s} ○ {si}; {si} ○ {s}	{si, eq, s}

Table 7. Compositions for Temporal Relation Reasoning

The following two examples are the rules for the relations “before” (i.e., {b}) and “before or meet or overlap” (i.e., {b, m, o}). More rules for temporal relation reasoning can be found Appendix 13.

```
INSERT INTO mdsys.semr_allenBasic VALUES(
'before2',
'(?t1 :overlap ?t2) (?t2 :before ?t3)',
null,
'(?t1 :before ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

```
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'b_m_o1',
'(?t1 :overlap ?t2) (?t2 :finished_by ?t3)',
null,
'(?t1 :before_or_meet_or_overlap ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

Based on these reasoning rules, all possible temporal relations specified in Table 5 can be found using Sem_Match () queries if they exist in the ontology. However, the inferred temporal relations between two time periods of antibiotic administration in the ontology maybe include more than one relation. For example, there are three antibiotic dose periods t1 (2010-10-05T08:00:00, 2010-10-07T10:30:00), t2 (2010-10-06T09:00:00, 2010-10-08T11:00:00) and t3 (2010-10-08T07:30:00, 2010-10-10T08:30:00). From the definitions of “before” and “overlap”, it can be inferred that t1 overlaps t2 (i.e., {o}), t2 overlaps t3 and t1 is before t3 (i.e., {b}). It can also be inferred from the composition {o} ○ {o} that t1 is before or meets or overlaps t3 (i.e., {b, m, o}). The inferred result about temporal relations between t1 and t3 is approximate. Therefore, it is important to determine which one is the exact temporal relation between the two time periods. If the inferred temporal relations between two time periods include a basic temporal relation, the basic relation should be an exact relation since all basic temporal relations are the minimal subsets in Allen’s temporal relations and are mutually exclusive to each other. In the previous example, the exact temporal relation between t1 and t3 is {b} since it is a basic relation. However, If the inferred relations between t1 and t3 are {bi, mi, oi, f, d} and {bi, mi, oi, si, di}, the exact relation between t1 and t3 should be {bi, mi, oi} which is the intersection of the previous two relations. Similarly, if the inferred relations between t1 and t3 are {b, m, o}, {b, m, o, s, d} and full, the exact relation between t1 and t3 should be {b, m, o} which is the minimal

subset in the intersections of these relations. Therefore, an indefinite fuzzy temporal relation is the exact relation between two time periods if the intersection of the inferred relations is an indefinite fuzzy temporal relation which is also the minimal subset in the intersections.

Finding the exact basic temporal relations between antibiotic dose periods in the ontology is straightforward since all basic relations are the exact relations. In order to find an indefinite fuzzy temporal relation which is an exact relation between two antibiotic administration periods, an extra set of reasoning rules are defined in the ontology which are used to determine the subset relationship between the 27 temporal relations in Table 5 and the intersection operation between these relations.

The Oracle rules below are used to determine the subset relationship between {b}, {b, m, o} and {b, m, o, fi, di}.

```
INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp1',
'(?r1 owl:equivalentProperty :before) (?r2 owl:equivalentProperty
:before_or_meet_or_overlap)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

```
INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp29',
'(?r1 owl:equivalentProperty :before_or_meet_or_overlap) (?r2
owl:equivalentProperty
:before_or_meet_or_overlap_or_finished_by_or_contain)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

```
INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp45',
'(?r1 :subSetOf ?r2) (?r2 :subSetOf ?r3)',
null,
'(?r1 :subSetOf ?r3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

The Oracle rules below are used to determine the intersection between {di, fi, o} and {di, oi, si}, and the intersection between {b, m, o, fi, di} and {b, m, o, s, d}.

```
INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
'relIntersection1',
'(?t1 :contain_or_finished_by_or_overlap ?t2) (?t1
:contain_or_overlapped_by_or_started_by ?t2)',
null,
'(?t1 :contain ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

```
INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
'relIntersection31',
'(?t1 :before_or_meet_or_overlap_or_finished_by_or_contain ?t2) (?t1
:before_or_meet_or_overlap_or_start_or_during ?t2)',
null,
'(?t1 :before_or_meet_or_overlap ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

There are total 45 rules defined for the subset relationship between temporal relations (Appendix 14) and 33 rules for the intersection between temporal relations (Appendix 15) in the ontology.

Based on these temporal relation definitions and reasoning rules about composition, subset relationship and intersection, Sem_Match () queries can be used to find all exact temporal relations between administered antibiotics for those patients. Suppose that there are some sepsis patients who took different antibiotics in different time periods. Some of dosing periods have both the start time and the end time charted whereas some of dosing periods only have one of them or none of them charted. There are some temporal relations between the dosing periods of these antibiotics which maybe are the definite basic relations or the indefinite fuzzy relations. In the following two Sem_Match () queries, the first query is for the basic relations and the second one is for the fuzzy relations which are represented by the abbreviations in the result due to the long string of their full names.

```
SELECT patient_label, administered_drugx$_suffix, ddx_startTime, ddx_endTime,
temporal_relation$_suffix, administered_drugy$_suffix, ddy_startTime, ddy_endTime
FROM TABLE(SEM_MATCH('{
?patient rdfs:label ?patient_label.
?patient :has_temporal_part ?patient_ddx_start.
?patient :has_temporal_part ?patient_ddx_end.
?patient_ddx_start :has_temporal_entity ?ddx_startTimeInstant.
?patient_ddx_end :has_temporal_entity ?ddx_endTimeInstant.
?administered_drugx :has_temporal_part ?drugx_ddx_start.
?administered_drugx :has_temporal_part ?drugx_ddx_end.
?drugx_ddx_start :has_temporal_entity ?ddx_startTimeInstant.
?drugx_ddx_end :has_temporal_entity ?ddx_endTimeInstant.
?patient_ddx_start :administered_with ?drugx_ddx_start.
?patient_ddx_end :administered_with ?drugx_ddx_end.
?patient :has_temporal_part ?patient_ddy_start.
?patient :has_temporal_part ?patient_ddy_end.
?patient_ddy_start :has_temporal_entity ?ddy_startTimeInstant.
?patient_ddy_end :has_temporal_entity ?ddy_endTimeInstant.
?administered_drugy :has_temporal_part ?drugy_ddy_start.
?administered_drugy :has_temporal_part ?drugy_ddy_end.
?drugy_ddy_start :has_temporal_entity ?ddy_startTimeInstant.
?drugy_ddy_end :has_temporal_entity ?ddy_endTimeInstant.
?patient_ddy_start :administered_with ?drugy_ddy_start.
?patient_ddy_end :administered_with ?drugy_ddy_end.
?ddx rdf:type :Time_Period. ?ddy rdf:type :Time_Period.
?ddx :open_instant ?ddx_startTimeInstant.
?ddx :close_instant ?ddx_endTimeInstant.
?ddy :open_instant ?ddy_startTimeInstant.
?ddy :close_instant ?ddy_endTimeInstant.

OPTIONAL { ?ddx_startTimeInstant :dateTimeValue ?ddx_startTime. }
OPTIONAL { ?ddx_endTimeInstant :dateTimeValue ?ddx_endTime. }
OPTIONAL { ?ddy_startTimeInstant :dateTimeValue ?ddy_startTime. }
OPTIONAL { ?ddy_endTimeInstant :dateTimeValue ?ddy_endTime. }

?ddx ?temporal_relation ?ddy.

FILTER (exists {?temporal_relation rdfs:subPropertyOf :allenBasic})
FILTER (EXISTS {?patient :present ?disease.
                    ?disease rdf:type :Sepsis_Uncertain_Focus. })  }',

SEM_MODELS('antibiotics'),
SEM_RULEBASES('owl2rl', 'allenBasic', 'allenFuzzy', 'allenRelationSubSetProp',
'allenRelationIntersection'),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')), null))
ORDER BY patient, administered_drugx, administered_drugy
```

PATIENT_LABEL	ADMINISTERED_DRUGX$_SUFFIX	DDX_STARTTIME	DDX_ENDTIME	TEMPORAL_RELATION$_SUFFIX	ADMINISTERED_DRUGY$_SUFFIX	DDY_STARTTIME	DDY_ENDTIME
Classie Murakami	flucloxacillin13	(null)	(null)	after	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00
Classie Murakami	flucloxacillin13	(null)	(null)	before	amikacin4	2010-01-20T08:00:00	(null)
Classie Murakami	flucloxacillin13	(null)	(null)	during	gentamicin28	(null)	(null)
Classie Murakami	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00	before	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00
Classie Murakami	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00	before	amikacin4	2010-01-20T08:00:00	(null)
Classie Murakami	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00	before	flucloxacillin12	(null)	(null)
Classie Murakami	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00	before	flucloxacillin13	(null)	(null)
Classie Murakami	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00	before	gentamicin28	(null)	(null)
Classie Murakami	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00	overlap	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00
Classie Murakami	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00	before	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00
Classie Murakami	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00	before	amikacin4	2010-01-20T08:00:00	(null)
Classie Murakami	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00	overlapped_by	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00
Classie Murakami	gentamicin28	(null)	(null)	after	flucloxacillin6	2010-01-10T14:00:00	2010-01-12T10:00:00
Classie Murakami	gentamicin28	(null)	(null)	before	amikacin4	2010-01-20T08:00:00	(null)
Classie Murakami	gentamicin28	(null)	(null)	contain	flucloxacillin13	(null)	(null)
Classie Murakami	gentamicin28	(null)	(null)	overlapped_by	flucloxacillin12	(null)	(null)
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	before	benzylpenicillin12	2010-01-11T03:00:00	(null)
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	start	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	start	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
David Brown	benzylpenicillin12	2010-01-11T03:00:00	(null)	after	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00
David Brown	benzylpenicillin12	2010-01-11T03:00:00	(null)	after	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00
David Brown	benzylpenicillin12	2010-01-11T03:00:00	(null)	met_by	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	before	benzylpenicillin12	2010-01-11T03:00:00	(null)
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	start	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	started_by	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00

Figure 26. Basic Temporal Relations between Administered Antibiotics

```
SELECT patient_label, administered_drugx$_suffix, ddx_startTime, ddx_endTime,
temporal_relation_lb, administered_drugy$_suffix, ddy_startTime, ddy_endTime
FROM TABLE(SEM_MATCH('{
?patient rdfs:label ?patient_label.
?patient :has_temporal_part ?patient_ddx_start. ?patient :has_temporal_part
?patient_ddx_end.
?patient_ddx_start :has_temporal_entity ?ddx_startTimeInstant.
?patient_ddx_end :has_temporal_entity ?ddx_endTimeInstant.
?administered_drugx :has_temporal_part ?drugx_ddx_start.
?administered_drugx :has_temporal_part ?drugx_ddx_end.
?drugx_ddx_start :has_temporal_entity ?ddx_startTimeInstant.
?drugx_ddx_end :has_temporal_entity ?ddx_endTimeInstant.
?patient_ddx_start :administered_with ?drugx_ddx_start.
?patient_ddx_end :administered_with ?drugx_ddx_end.
?patient :has_temporal_part ?patient_ddy_start.
?patient :has_temporal_part ?patient_ddy_end.
?patient_ddy_start :has_temporal_entity ?ddy_startTimeInstant.
?patient_ddy_end :has_temporal_entity ?ddy_endTimeInstant.
?administered_drugy :has_temporal_part ?drugy_ddy_start.
?administered_drugy :has_temporal_part ?drugy_ddy_end.
?drugy_ddy_start :has_temporal_entity ?ddy_startTimeInstant.
?drugy_ddy_end :has_temporal_entity ?ddy_endTimeInstant.
?patient_ddy_start :administered_with ?drugy_ddy_start.
?patient_ddy_end :administered_with ?drugy_ddy_end.
?ddx rdf:type :Time_Period. ?ddy rdf:type :Time_Period.
?ddx :open_instant ?ddx_startTimeInstant. ?ddx :close_instant ?ddx_endTimeInstant.
?ddy :open_instant ?ddy_startTimeInstant. ?ddy :close_instant ?ddy_endTimeInstant.

OPTIONAL { ?ddx_startTimeInstant :dateTimeValue ?ddx_startTime. }
OPTIONAL { ?ddx_endTimeInstant :dateTimeValue ?ddx_endTime. }
OPTIONAL { ?ddy_startTimeInstant :dateTimeValue ?ddy_startTime. }
OPTIONAL { ?ddy_endTimeInstant :dateTimeValue ?ddy_endTime. }

?temporal_relation rdfs:label ?temporal_relation_lb. ?ddx ?temporal_relation ?ddy.

FILTER (?ddx != ?ddy)
FILTER (exists {?temporal_relation rdfs:subPropertyOf :allenFuzzy})
FILTER (not exists {?ddx ?anotherTempRel ?ddy.
                    ? anotherTempRel :subSetOf ?temporal_relation})
FILTER (EXISTS {?patient :present ?disease.
                ?disease rdf:type :Sepsis_Uncertain_Focus. }) }',
SEM_MODELS('antibiotics'),
SEM_RULEBASES('owl2rl', 'allenBasic', 'allenFuzzy', 'allenRelationSubSetProp',
'allenRelationIntersection),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')), null))
ORDER BY patient, administered_drugx, administered_drugy
```

	PATIENT_LABEL	ADMINISTERED_DRUGX$_SUFFIX	DDX_STARTTIME	DDX_ENDTIME	TEMPORAL_RELATION_LB	ADMINISTERED_DRUGY$_SUFFIX	DDY_STARTTIME	DDY_ENDTIME
1	Classie Murakami	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00	full	flucloxacillin12	(null)	(null)
2	Classie Murakami	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00	full	flucloxacillin13	(null)	(null)
3	Classie Murakami	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00	full	gentamicin28	(null)	(null)
4	Classie Murakami	flucloxacillin12	(null)	(null)	full	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00
5	Classie Murakami	flucloxacillin12	(null)	(null)	b_m_o_fi_di	flucloxacillin13	(null)	(null)
6	Classie Murakami	flucloxacillin12	(null)	(null)	bi_mi_oi_f_d	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00
7	Classie Murakami	flucloxacillin13	(null)	(null)	full	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00
8	Classie Murakami	flucloxacillin13	(null)	(null)	bi_mi_oi_f_d	flucloxacillin12	(null)	(null)
9	Classie Murakami	flucloxacillin13	(null)	(null)	bi_mi_oi_f_d	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00
10	Classie Murakami	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00	b_m_o_fi_di	flucloxacillin12	(null)	(null)
11	Classie Murakami	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00	b_m_o_fi_di	flucloxacillin13	(null)	(null)
12	Classie Murakami	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00	b_m_o_fi_di	gentamicin28	(null)	(null)
13	Classie Murakami	gentamicin28	(null)	(null)	full	amikacin1	2010-01-13T16:00:00	2010-01-15T08:00:00
14	Classie Murakami	gentamicin28	(null)	(null)	bi_mi_oi_f_d	gentamicin17	2010-01-10T16:00:00	2010-01-12T16:00:00
15	David Brown	benzylpenicillin12	2010-01-11T03:00:00	(null)	full	flucloxacillin14	(null)	2010-01-12T08:00:00
16	David Brown	flucloxacillin14	(null)	2010-01-12T08:00:00	full	benzylpenicillin12	2010-01-11T03:00:00	(null)
17	David Brown	flucloxacillin14	(null)	2010-01-12T08:00:00	bi_mi_oi	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
18	David Brown	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00	b_m_o	flucloxacillin14	(null)	2010-01-12T08:00:00
19	Lucy Bake	flucloxacillin3	2010-01-08T08:00:00	2010-01-11T00:00:00	b_m_o_s_d	vancomycin33	(null)	(null)
20	Lucy Bake	gentamicin3	2010-01-08T08:00:00	2010-01-12T21:00:00	b_m_o_s_d	vancomycin33	(null)	(null)
21	Lucy Bake	vancomycin2	2010-01-08T10:00:00	2010-01-10T22:00:00	b_m_o_s_d	vancomycin33	(null)	(null)
22	Lucy Bake	vancomycin33	(null)	(null)	bi_mi_oi_si_di	flucloxacillin3	2010-01-08T08:00:00	2010-01-11T00:00:00
23	Lucy Bake	vancomycin33	(null)	(null)	bi_mi_oi_si_di	gentamicin3	2010-01-08T08:00:00	2010-01-12T21:00:00
24	Lucy Bake	vancomycin33	(null)	(null)	bi_mi_oi_si_di	vancomycin2	2010-01-08T10:00:00	2010-01-10T22:00:00
25	Michael Jones	flucloxacillin4	2010-01-08T12:00:00	2010-01-10T14:00:00	b_m_o_fi_di	spectinomycin2	(null)	(null)
26	Michael Jones	gentamicin4	2010-01-08T14:00:00	2010-01-11T14:00:00	b_m_o_fi_di	spectinomycin2	(null)	(null)
27	Michael Jones	spectinomycin2	(null)	(null)	bi_mi_oi_f_d	flucloxacillin4	2010-01-08T12:00:00	2010-01-10T14:00:00
28	Michael Jones	spectinomycin2	(null)	(null)	bi_mi_oi_f_d	gentamicin4	2010-01-08T14:00:00	2010-01-11T14:00:00

Figure 27. Fuzzy Temporal Relations between Administered Drugs

5.2.3.4 Temporal Relation Inconsistency Checking in Administered Antibiotics

Finding inconsistent temporal relations in the ontology is another important task in temporal reasoning. Inconsistency is caused by temporal relations between the same two events which conflict with each other. In other words, these relations connecting the same two events are disjoint with each other in the ontology. From the point of view of set operation as shown below, if two relations R_1 and R_2 exist between two events X and Y, and the intersection of the two relation sets yields an empty set, then the temporal relations between the two events are not consistent.

$R_1 \cap R_2 \rightarrow \emptyset$

For example, there are two temporal relations {b} and {m} between the events X and Y. The following intersection of {b} and {m} leads to an empty set since the relations “b” and “m” are mutually exclusive (i.e., disjoint) in Allen’s algebra.

$\{b\} \cap \{m\} \rightarrow \emptyset$

These two relations {b, m, o} and {s} between the events X and Y also conflict with each other. However, these two relations {d, f, oi} and {d} do not conflict with each other since the intersection of {d, f, oi} and {d} yields a result set {d}.

Inconsistency can occur between a basic relation and another basic relation, a basic relation and a fuzzy relation, or a fuzzy relation and another fuzzy relation. There are 223 disjoint relation pairs (Table 8) for the 27 temporal relations in Table 5.

Disjointness between basic relation and basic relation (78)
<{b}, {bi}>, <{b}, {m}>, <{b}, {mi}>, <{b}, {o}>, <{b}, {oi}>, <{b}, {s}>, <{b}, {si}>, <{b}, {eq}>, <{b}, {d}>, <{b}, {di}>, <{b}, {f}>, <{b}, {fi}>, <{bi}, {m}>, <{bi}, {mi}>, <{bi}, {o}>, <{bi}, {oi}>, <{bi}, {s}>, <{bi}, {si}>, <{bi}, {eq}>, <{bi}, {d}>, <{bi}, {di}>, <{bi}, {f}>, <{bi}, {fi}>, <{m}, {mi}>, <{m}, {o}>, <{m}, {oi}>, <{m}, {s}>, <{m}, {si}>, <{m}, {eq}>, <{m}, {d}>, <{m}, {di}>, <{m}, {f}>, <{m}, {fi}>, <{mi}, {o}>, <{mi}, {oi}>, <{mi}, {s}>, <{mi}, {si}>, <{mi}, {eq}>, <{mi}, {d}>, <{mi}, {di}>, <{mi}, {f}>, <{mi}, {fi}>, <{o}, {oi}>, <{o}, {s}>, <{o}, {si}>, <{o}, {eq}>, <{o}, {d}>, <{o}, {di}>,<{o}, {f}>, <{o}, {fi}>, <{oi}, {s}>, <{oi}, {si}>, <{oi}, {eq}>, <{oi}, {d}>, <{oi}, {di}>, <{oi}, {f}>, <{oi}, {fi}>, <{s}, {si}>, <{s}, {eq}>, <{s}, {d}>, <{s}, {di}>, <{s}, {f}>, <{s}, {fi}>, <{si}, {eq}>, <{si}, {d}>, <{si}, {di}>, <{si}, {f}>, <{si}, {fi}>, <{eq}, {d}>, <{eq}, {di}>, <{eq}, {f}>, <{eq}, {fi}>, <{d}, {di}>, <{d}, {f}>, <{d}, {fi}>, <{di}, {f}>, <{di}, {fi}>, <{f}, {fi}>
Disjointness between basic relation and fuzzy relation (116)
<{b}, {bi, mi, oi}>, <{b}, {bi, mi, oi, f, d}>, <{b}, {bi, mi, oi, si, di}>, <{b}, concur>, <{b}, {d, f, oi}>, <{b}, {di, fi, o}>, <{b}, {d, o, s}>, <{b}, {di, oi, si}>, <{b}, {f, eq, fi}>, <{b}, {si, eq, s}>, <{bi}, {b, m, o}>, <{bi}, {b, m, o, fi, di}>, <{bi}, {b, m, o, s, d}>, <{bi}, concur>, <{bi}, {d, f, oi}>, <{bi}, {di, fi, o}>, <{bi}, {d, o, s}>, <{bi}, {di, oi, si}>, <{bi}, {f, eq, fi}>, <{bi}, {si, eq, s}>, <{m}, {bi, mi, oi}>, <{m}, {bi, mi, oi, f, d}>, <{m}, {bi, mi, oi, si, di}>, <{m}, concur>, <{m}, {d, f, oi}>, <{m}, {di, fi, o}>, <{m}, {d, o, s}>, <{m}, {di, oi, si}>, <{m}, {f, eq, fi}>, <{m}, {si, eq, s}>, <{mi}, {b, m, o}>, <{mi}, {b, m, o, fi, di}>, <{mi}, {b, m, o, s, d}>, <{mi}, concur>, <{mi}, {d, f, oi}>, <{mi}, {di, fi, o}>, <{mi}, {d, o, s}>, <{mi}, {di, oi, si}>, <{mi}, {f, eq, fi}>, <{mi}, {si, eq, s}>, <{o}, {bi, mi, oi}>, <{o}, {bi, mi, oi, f, d}>, <{o}, {bi, mi, oi, si, di}>, <{o}, {d, f, oi}>, <{o}, {di, oi, si}>, <{o}, {f, eq, fi}>, <{o}, {si, eq, s}>, <{oi}, {b, m, o}>, <{oi}, {b, m, o, fi, di}>, <{oi}, {b, m, o, s, d}>, <{oi}, {di, fi, o}>, <{oi}, {d, o, s}>, <{oi}, {f, eq, fi}>, <{oi}, {si, eq, s}>, <{s}, {b, m, o}>, <{s}, {bi, mi, oi}>, <{s}, {b, m, o, fi, di}>, <{s}, {bi, mi, oi, f, d}>, <{s}, {bi, mi, oi, si, di}>, <{s}, {d, f, oi}>, <{s}, {di, fi, o}>, <{s}, {di, oi, si}>, <{s}, {f, eq, fi}>, <{si}, {b, m, o}>, <{si}, {bi, mi, oi}>, <{si}, {b, m, o, fi, di}>, <{si}, {bi, mi, oi, f, d}>, <{si}, {b, m, o, s, d}>, <{si}, {d, f, oi}>, <{si}, {di, fi, o}>, <{si}, {d, o, s}>, <{si}, {f, eq, fi}>, <{eq}, {b, m, o}>, <{eq}, {bi, mi, oi}>, <{eq}, {b, m, o, fi, di}>, <{eq}, {bi, mi, oi, f, d}>, <{eq}, {b, m, o, s, d}>, <{eq}, {bi, mi, oi, si, di}>, <{eq}, {d, f, oi}>, <{eq}, {di, fi, o}>, <{eq}, {d, o, s}>, <{eq}, {di, oi, si}>, <{d}, {b, m, o}>, <{d}, {bi, mi, oi}>, <{d}, {b, m, o, fi, di}>, <{d}, {bi, mi, oi, si, di}>, <{d}, {di, fi, o}>, <{d}, {di, oi, si}>, <{d}, {si, eq, s}>, <{d}, {f, eq, fi}>, <{di}, {b, m, o}>, <{di}, {bi, mi, oi}>, <{di}, {bi, mi, oi, f, d}>, <{di}, {b, m, o, s, d}>, <{di}, {d, f, oi}>, <{di}, {d, o, s}>, <{di}, {si, eq, s}>, <{di}, {f, eq, fi}>, <{f}, {b, m, o}>, <{f}, {bi, mi, oi}>, <{f}, {b, m, o, fi, di}>, <{f}, {b, m, o, s, d}>, <{f}, {bi, mi, oi, si, di}>, <{f}, {di, fi, o}>, <{f}, {d, o, s}>, <{f}, {di, oi, si}>, <{f}, {si, eq, s}>, <{fi}, {b, m, o}>, <{fi}, {bi, mi, oi}>, <{fi}, {bi, mi, oi, f, d}>, <{fi}, {b, m, o, s, d}>, <{fi}, {bi, mi, oi, si, di}>, <{fi}, {d, f, oi}>, <{fi}, {d, o, s}>, <{fi}, {di, oi, si}>, <{fi}, {si, eq, s}>
Disjointness between fuzzy relation and fuzzy relation (29)
<{b, m, o}, {bi, mi, oi}>, <{b, m, o}, {bi, mi, oi, f, d}>, <{b, m, o}, {bi, mi, oi, si, di}>, <{b, m, o}, {d, f, oi}>, <{b, m, o}, {di, oi, si}>, <{b, m, o}, {f, eq, fi}>, <{b, m, o}, {si, eq, s}>, <{bi, mi, oi}, {b, m, o, fi, di}>, <{bi, mi, oi}, {b, m, o, s, d}>, <{bi, mi, oi}, {di, fi, o}>, <{bi, mi, oi}, {d, o, s}>, <{bi, mi, oi}, {f, eq, fi}>, <{bi, mi, oi}, {si, eq, s}>, <{b, m, o, fi, di}, {bi, mi, oi, f, d}>, <{b, m, o, fi, di}, {d, f, oi}>, <{b, m, o, fi, di}, {si, eq, s}>, <{bi, mi, oi f, d}, {di, fi, o}>, <{bi, mi, oi, f, d}, {si, eq, s}>, <{b, m, o, s, d}, {bi, mi, oi, si, di}>, <{b, m, o, s, d}, {di, oi, si}>, <{b, m, o, s, d}, {f, eq, fi}>, <{bi, mi, oi, si, di}, {d, o, s}>, <{bi, mi, oi, si, di}, {f, eq ,fi}>, <{d, f, oi}, {di, fi, o}>, <{d, f, oi}, {si, eq, s}>, <{di, fi, o}, {si, eq, s}>, <{d, o, s}, {di, oi, si}>, <{d, o, s}, {f, eq, fi}>, <{di, oi, si}, {f,eq,fi}>
Total: 223

Table 8. Pair of Disjoint Temporal Relations

In order to check all possible inconsistent temporal relations for these relations, it will need 223 temporal reasoning rules to be defined using Oracle user defined rules. However, it is very tedious and error prone to manually write these rules. In this antibiotic treatment guideline ontology, an Oracle user-defined inference function is developed for implementing the inconsistency checking task. This function contains only 23 SQL queries (Appendix 16) for checking all possible inconsistent temporal relations in the ontology. Each SQL query contained in this function will find all disjoint relations for a particular temporal relation. For example, the following SQL query in this function will find all disjoint relations of “before” relation which are the rest 12 basic relations, and the indefinite fuzzy relations {bi, mi, oi}, {bi, mi, oi, f, d}, {bi, mi, oi, si, di}, concur, {d, f, oi}, {di, fi, o}, {d, o, s}, {di, oi, si}, {f, eq, fi} and {si, eq, s}.

```
sqlStmt1 := 'SELECT ids1.sid dose1, ids2.sid dose2, ids3.pid timeRel,
ids4.pid conflictTimeRel
FROM
' || src_tab_view || ' ids1, ' || src_tab_view || ' ids2,
' || src_tab_view || ' ids3, ' || src_tab_view || ' ids4
WHERE ids1.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND ids1.oid = ' || to_char(timePeriodClassId,'TM9') || '
AND ids2.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND ids2.oid = ' || to_char(timePeriodClassId,'TM9') || '

AND ids3.sid = ids1.sid
AND ids3.pid = ' || to_char(beforePropertyId ,'TM9') || '
AND ids3.oid = ids2.sid
AND ids4.sid = ids1.sid

AND ((ids4.pid = '|| to_char(afterPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(meetPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(metByPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(overlapPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(overlappedByPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(startPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(startedByPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(equalPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(duringPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(containPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(finishPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(finishedByPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(bi_mi_oiPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(bi_mi_oi_f_dPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(bi_mi_oi_si_diPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(concurPropertyId,'TM9') || ') OR
```

```
    (ids4.pid = '|| to_char(d_f_oiPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(di_fi_oPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(d_o_sPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(di_oi_siPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(f_eq_fiPropertyId,'TM9') || ') OR
    (ids4.pid = '|| to_char(si_eq_sPropertyId,'TM9') || ')
    )
AND ids4.oid = ids2.sid ';

insertStmt1 := 'INSERT  INTO ' || output_tab || ' (sid, pid, oid)
SELECT timeRel, ' || to_char(confilctPropertyId,'TM9') || ', conflictTimeRel
FROM (' || sqlStmt1 || ')
UNION
SELECT dose1, ' || to_char(hasConfRelPropertyId,'TM9') || ',dose2
FROM (' || sqlStmt1 || ')
';
```

Suppose that a new temporal relation “overlap” between administered antibiotics benzylpenicillin12 and flucloxacillin5 for the patient David Brown is added into the ontology which might lead to inconsistent temporal relations between these two drugs. Based on that function and the extended 4D fluent representation method, the following Sem_Match () query is able to find the conflict relations between the two drugs which are caused by the new added relation. The result is shown in Figure 28.

The returned non-empty result in Figure 28 implies there are some temporal relations in the ontology which conflict with each other and need to be corrected. By running the inconsistency checking function from time to time, the consistency of temporal relations in the ontology can be ensured such that the results from queries of temporal relations between administered drugs can be correctly returned to clinicians.

```
SELECT patient_label, administered_drugx$_suffix, temporal_relation_lb,
conflict_relation_lb, administered_drugy$_suffix
FROM TABLE(SEM_MATCH('{ ?patient rdfs:label ?patient_label.
?patient :has_temporal_part ?patient_ddx_start.
?patient :has_temporal_part ?patient_ddx_end.
?patient_ddx_start :has_temporal_entity ?ddx_startTimeInstant.
?patient_ddx_end :has_temporal_entity ?ddx_endTimeInstant. ?administered_drugx
:has_temporal_part ?drugx_ddx_start.
?administered_drugx :has_temporal_part ?drugx_ddx_end.
?drugx_ddx_start :has_temporal_entity ?ddx_startTimeInstant.
?drugx_ddx_end :has_temporal_entity ?ddx_endTimeInstant.
?patient_ddx_start :administered_with ?drugx_ddx_start.
?patient_ddx_end :administered_with ?drugx_ddx_end.
?patient :has_temporal_part ?patient_ddy_start.
?patient :has_temporal_part ?patient_ddy_end.
?patient_ddy_start :has_temporal_entity ?ddy_startTimeInstant.
?patient_ddy_end :has_temporal_entity ?ddy_endTimeInstant. ?administered_drugy
:has_temporal_part ?drugy_ddy_start.
?administered_drugy :has_temporal_part ?drugy_ddy_end.
?drugy_ddy_start :has_temporal_entity ?ddy_startTimeInstant.
?drugy_ddy_end :has_temporal_entity ?ddy_endTimeInstant.
?patient_ddy_start :administered_with ?drugy_ddy_start.
?patient_ddy_end :administered_with ?drugy_ddy_end.

?dose_periodx rdf:type :Time_Period. ?dose_periody rdf:type :Time_Period.
?dose_periodx :open_instant ?ddx_startTimeInstant.
?dose_periodx :close_instant ?ddx_endTimeInstant.
?dose_periody :open_instant ?ddy_startTimeInstant.
?dose_periody :close_instant ?ddy_endTimeInstant.
?temporal_relation rdfs:label ?temporal_relation_lb.
?conflict_relation rdfs:label ?conflict_relation_lb.
?dose_periodx ?temporal_relation ?dose_periody.
?dose_periodx ?conflict_relation ?dose_periody.

FILTER((?administered_drugx=:benzylpenicillin12)&&(?administered_drugy=:flucloxacillin5))
FILTER ((?temporal_relation != :allenBasic) &&
(?temporal_relation != :has_conflict_temporal_relation) &&
(?temporal_relation != :allenFuzzy) && (?conflict_relation != :allenBasic) &&
(?conflict_relation != :has_conflict_temporal_relation) &&
(?conflict_relation != :allenFuzzy) )
FILTER (EXISTS { ?temporal_relation :conflict ?conflict_relation }) } ',
SEM_Models('antibiotics'), SEM_RULEBASES('owl2rl', 'allenBasic', ‘allenFuzzy’,
'allenRelationSubSetProp', 'allenRelationIntersection'),
SEM_ALIASES(SEM_ALIAS('', 'http://www.usyd.edu.au/hitru/antibiotics#')), null))
ORDER BY patient, administered_drugx, administered_drugy
```

	PATIENT_LABEL	ADMINISTERED_DRUGX$_SUFFIX	TEMPORAL_RELATION$_SUFFIX	CONFLICT_RELATION$_SUFFIX	ADMINISTERED_DRUGY$_SUFFIX
1	David Brown	benzylpenicillin12	after	contain	flucloxacillin5
2	David Brown	benzylpenicillin12	after	during	flucloxacillin5
3	David Brown	benzylpenicillin12	after	overlap	flucloxacillin5
4	David Brown	benzylpenicillin12	after	overlapped_by	flucloxacillin5
5	David Brown	benzylpenicillin12	before	after	flucloxacillin5
6	David Brown	benzylpenicillin12	before	contain	flucloxacillin5
7	David Brown	benzylpenicillin12	before	during	flucloxacillin5
8	David Brown	benzylpenicillin12	before	overlap	flucloxacillin5
9	David Brown	benzylpenicillin12	before	overlapped_by	flucloxacillin5
10	David Brown	benzylpenicillin12	during	contain	flucloxacillin5
11	David Brown	benzylpenicillin12	overlap	contain	flucloxacillin5
12	David Brown	benzylpenicillin12	overlap	during	flucloxacillin5
13	David Brown	benzylpenicillin12	overlap	overlapped_by	flucloxacillin5
14	David Brown	benzylpenicillin12	overlapped_by	contain	flucloxacillin5
15	David Brown	benzylpenicillin12	overlapped_by	during	flucloxacillin5

Figure 28. Inconsistent Temporal Relations between Administered Antibiotics of Patient David Brown

5.3 System Architecture

A prototype ontology system for the QUAIC antibiotic treatment guidelines is also developed in Java using Jena Adapter. The Jena Adapter provides a Java based interface to the guideline ontology stored in Oracle RDF Semantic Graph. The system architecture is shown in Figure 29.

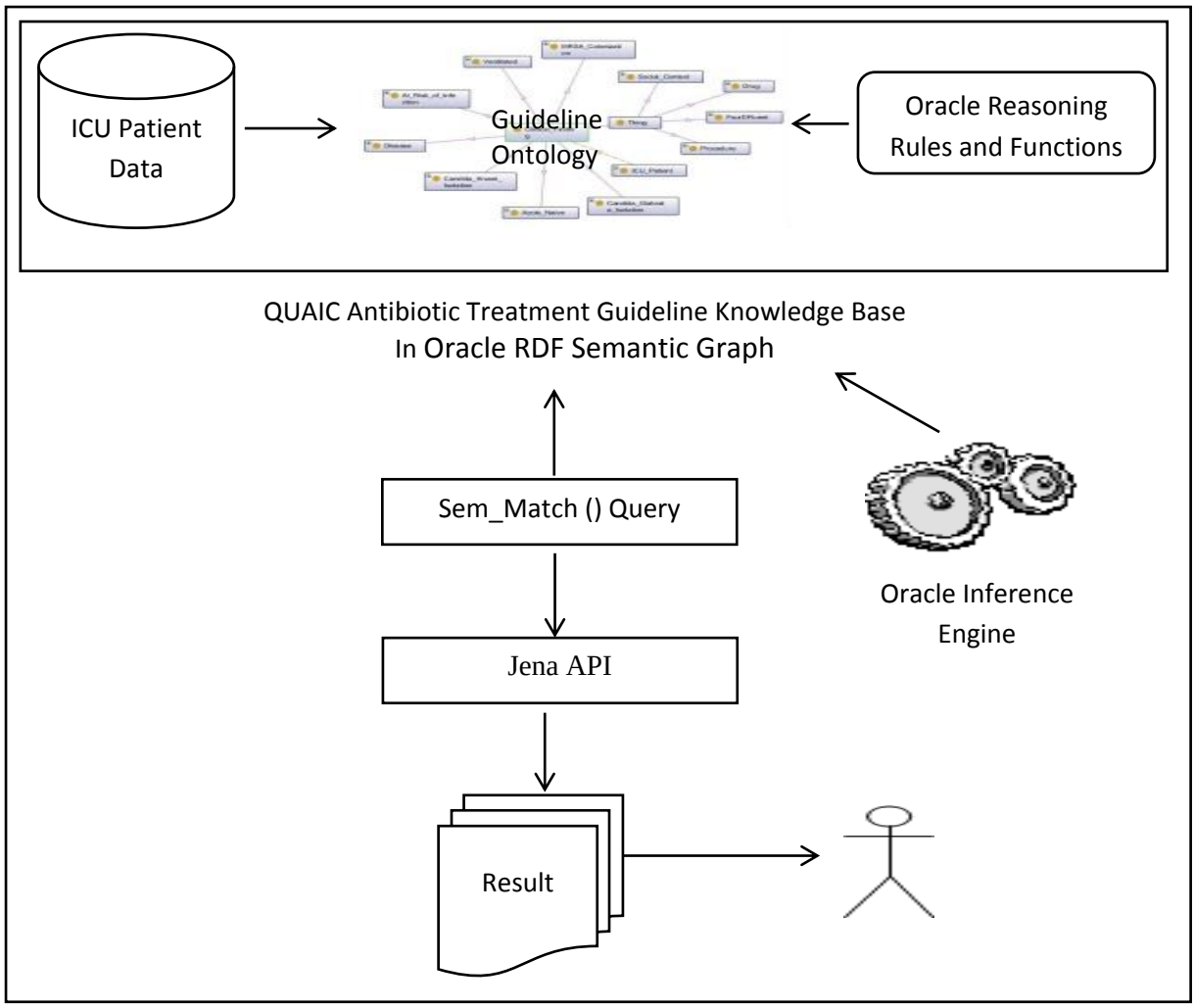


Figure 29. System Architecture of the Antibiotic Treatment Guideline Ontology System

The prototype system has a navigation menu (Figure 30) to assist end users to find recommended regimens, administered antibiotics, dose interval, dose duration and temporal relations between administered antibiotics. If a user wants to find those information for ICU patients in a particular category such as sepsis (uncertain focus), he or she needs to select one choice from the navigation menu. The returned result is displayed in a graphical interface (Figure 31).

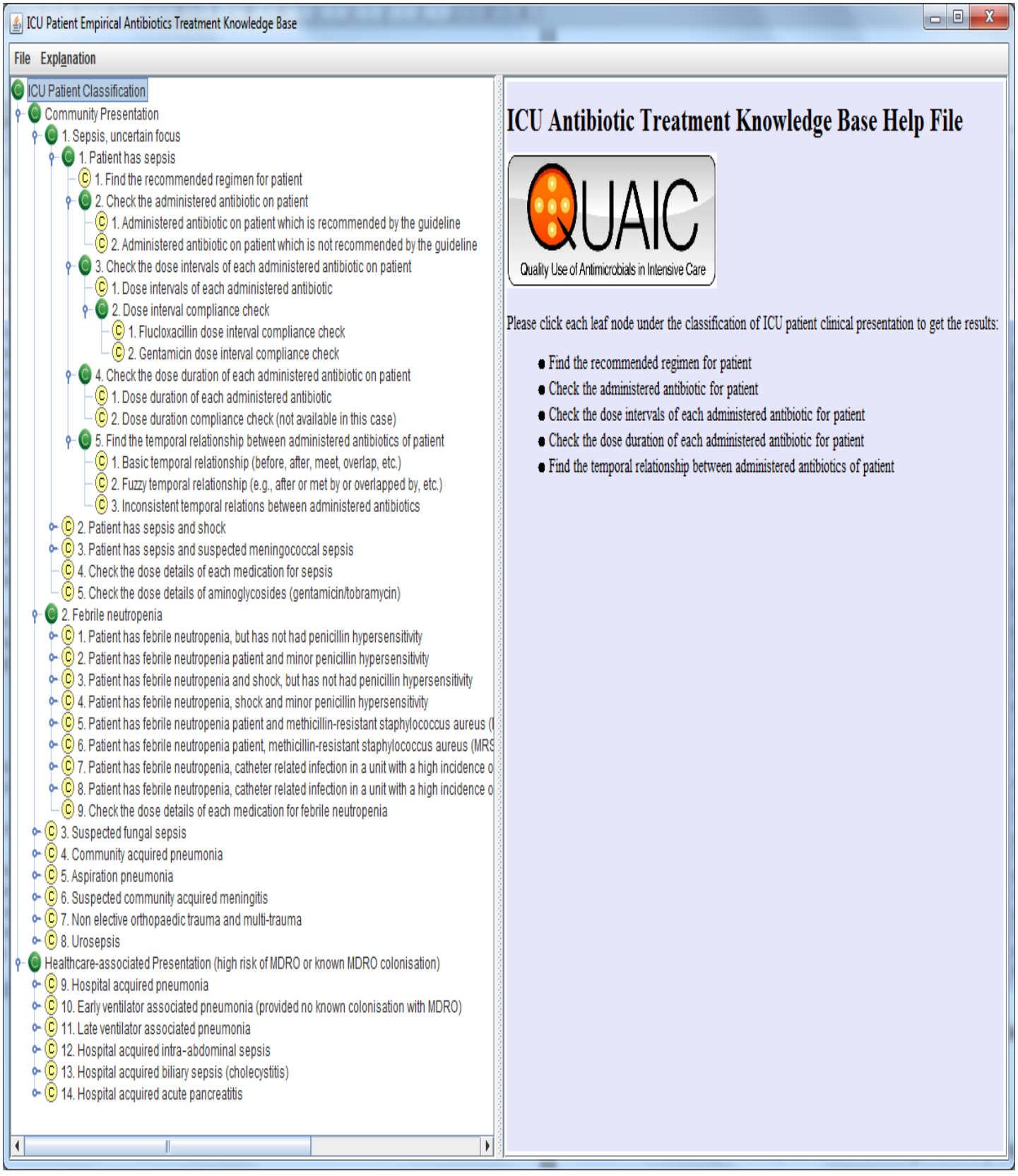


Figure 30. System Navigation Menu

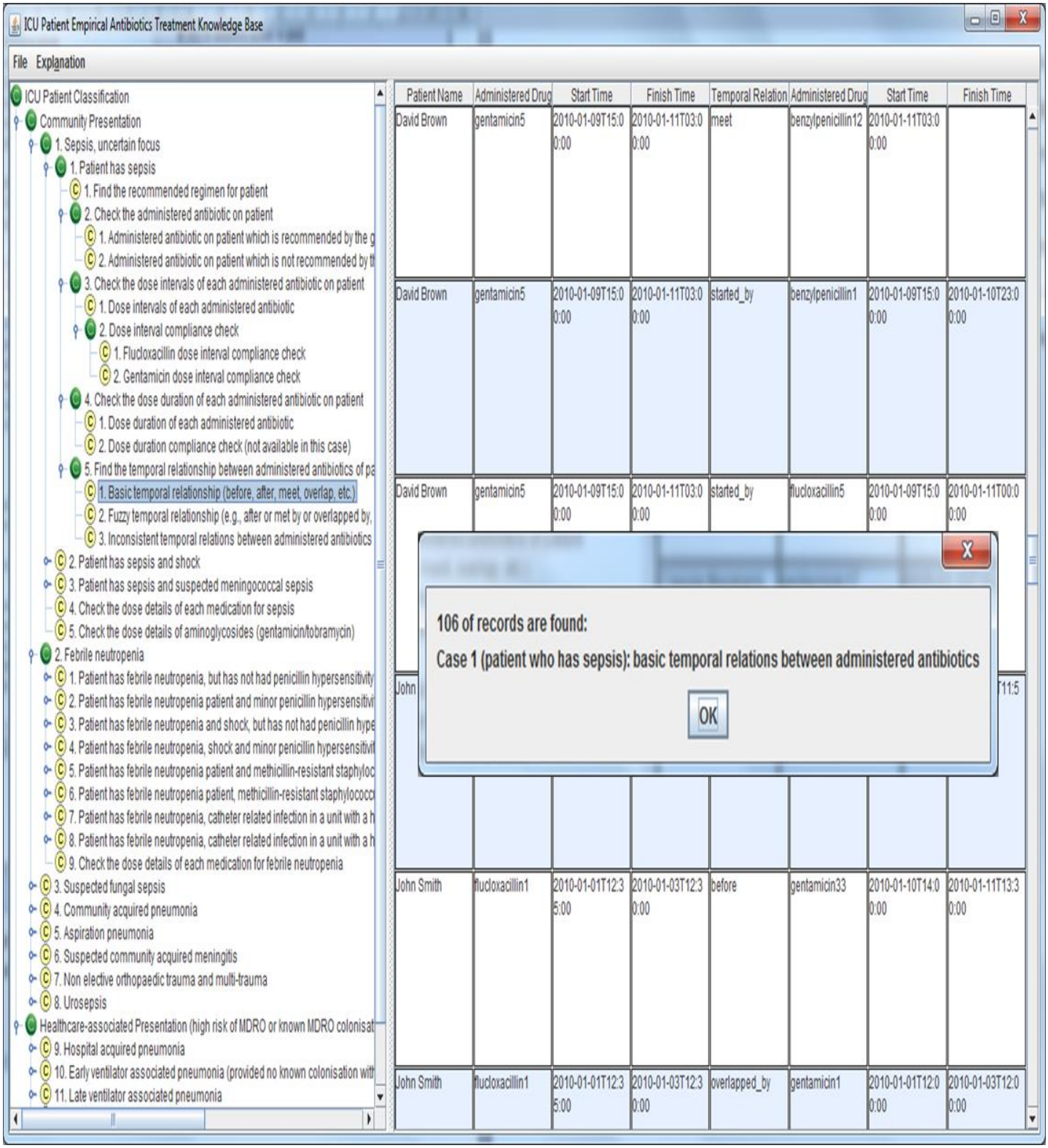


Figure 31. Sample Output in a Graphical Interface

In summary, clinical knowledge reasoning, especially temporal knowledge reasoning, for the antibiotic treatment guideline ontology is analysed in this chapter. The analysis shows that clinical knowledge reasoning to find the recommended regimens for ICU patients can be achieved by ordinary Oracle user-defined rules and inference functions. However, the implementation of temporal knowledge reasoning in the ontology needs a temporal knowledge modelling method which is the extended 4D fluent temporal knowledge modelling method in this project. The extended 4D fluent method is demonstrated in the QUAIC antibiotic treatment guideline ontology to represent temporal knowledge contained in regimen recommendations. The extended 4D fluent representation method not only enables the finding of administered antibiotics, dose interval and dose duration of administered antibiotics, but also enables the finding the temporal relations between administered drugs by leveraging Allen's interval algebra.

Chapter 6 Evaluation

The evaluation of the extended 4D fluent approach on the QUAIC antibiotic treatment guideline ontology has two aspects: the evaluation of logic consistency of the ontology and the evaluation of clinical question answering in the ontology based on patient data.

6.1 Evaluation of Logic Consistency of the Ontology

From the point of view of formal semantics, an ontology is defined as a pair $O = (T, A)$ where T denotes the TBox containing terminology axioms and role axioms in the ontology, and A denotes the ABox containing assertional axioms in the ontology. The semantics of the ontology O is defined by an interpretation function $I(\Delta^I, \cdot^I)$ where Δ^I denotes a non-empty set domain and $\cdot^I$ denotes the interpretation function. The interpretation function maps individuals, concepts and roles to elements of the domain, subsets of the domain and binary relations on the domain, respectively. This terminology axiom $C \sqsubseteq D$ can be satisfied by an interpretation if $C^I \sqsubseteq D^I$. Similarly, this assertional axiom $C(a)$ can be satisfied by an interpretation if $a^I \in C^I$. An interpretation is called a model of the ontology if and only if it satisfies each axiom in that ontology. Thus, an inconsistency in an ontology O means there is no a model that can satisfy each axiom in O [57]. In other words, an inconsistency in an ontology means that ontology contains one or more axioms which are logically contradictory. An inconsistent ontology prevents useful information to be inferred from the ontology in an OWL reasoner. Therefore, ensuring the consistency of the ontology is a necessary step for the evaluation of clinical question answering in the ontology.

The purpose of evaluating the logical consistency of the ontology is to find any axioms in the TBox and ABox of the ontology which are potentially logically contradictory. Based on the logical inconsistencies of ontology discussed in [57] and [58], the evaluation of the antibiotic treatment guideline ontology is based on the following inconsistency types via the Protégé built-in reasoners Pellet, FaCT++ and HermiT. Eight different types of inconsistency can be identified and the evaluation result is summarised in Table 9.

1. An unsatisfiable class in the ontology. A class is unsatisfiable if and only if the interpretation of the class in the ontology is empty with regard to each model of

the ontology. That is to say an unsatisfiable class cannot have instances in the ontology. For example, class C is unsatisfiable if C is the subclass of both class A and class B where A and B are disjoint with each other or complement of each other. If an individual is created to initiate C, it will lead to an inconsistency error in the ontology. Unsatisfiable classes need to be removed from the ontology to avoid this type of inconsistency error.

2. Disjoint classes with same individuals. Two classes are said to be disjoint with each other if no individual can be the member of both classes at the same time. For example, if class A is disjoint with class B and an individual is created to be the member of both A and B, it will lead to an inconsistent ontology. Disjoint relation between superclasses is inherited by their subclasses.
3. Complement classes with same individuals. Class A is a complement of another class B if it contains all individuals which are not the members of B. If an individual is initiated to be the member of both A and B, it will lead to an inconsistent ontology.
4. Disjoint classes sharing nominal classes. A nominal class is defined to be the enumeration of individuals. For example, class A is defined to be equivalent to {x, y, z} which is the enumeration of individuals x, y and z. If a nominal class is defined to be a subclass of disjoint classes, it will lead to an inconsistent ontology.
5. Disjoint object properties connecting same individuals. Similarly to disjoint classes, same individuals cannot be connected by object properties which are disjoint. For example, there are two individuals x and y, and two object properties property1 and property2 where property1 is disjoint with property2. If x is connected to y by both property1 and property2, it will lead to an inconsistent ontology.
6. Disjoint data properties for same individuals. If two data properties are disjoint with each other, an individual cannot have these two properties with same data type value. For example, two data properties property1 and property2 are disjoint with each other. For an individual x, x cannot have both property1 and property2 with a same data type value.

7. Cardinality restriction on object properties and data properties. If the number of individuals connected by a property violates the cardinality restriction on that property, it will lead to an inconsistent ontology. For example, the cardinality restriction on “hasWife” object property is maximum 1 and Peter has wives who are Mary and Lucy. If Mary and Lucy are not explicitly stated as the same person in the ontology, it will violate the maximum cardinality restriction on that property and lead to an inconsistent ontology.
8. Datatype range restriction on data properties. If a literal value violates the range restriction on a data property, it will lead to an inconsistent ontology. For example, the restriction on “dateOfBirth” data property is the dateTime data type. If the actual value for that property is set to a string, it will violate the restriction and lead to an inconsistent ontology.

Types of Inconsistency	**Result of Inconsistency checking via Pellet, FaCT++ and HermiT**
1	No unsatisfiable classes are found in total 172 classes of the ontology.
2	In total 946 pairs of asserted and inferred disjoint classes in the ontology, none of them contains same individuals.
3	No complement classes are found in total 172 classes of the ontology. Therefore, no complement classes with same individuals are found in the ontology.
4	No disjoint classes sharing nominal classes are found in total 172 classes of the ontology.
5	No disjoint object properties connecting same individuals are found in total 36 object properties of the ontology.
6	No disjoint data properties for same individuals are found in total 16 data properties of the ontology.
7	No violations of cardinality restrictions on object and data properties are found in the ontology.
8	No violations of datatype range restrictions on total 16 data properties of the ontology are found.

Table 9. Summary of Logical Inconsistency Checking in the Ontology

6.2 Evaluation of Clinical Question Answering

As analysed in Chapter 5, there are five major clinical questions (Table 10) which the reasoning rules and functions of the ontology can answer. Among the five clinical questions, four of them rely on the extended 4D fluent method and are the focus of the evaluation. The purpose of evaluating clinical question answering in the ontology is to validate if these rules and functions can give the expected output from the input of relevant patient information in terms of each medical case.

No. of Clinical Question		Clinical Question
Questions which are not based on the extended 4D fluent method	Q1	What are the recommended antibiotic regimens for a patient if the patient has the clinical presentation described in the guideline such as sepsis and pneumonia etcetera?
Questions which are based on the extended 4D fluent method	Q2	What are the administered antibiotics for a patient who has recommendations and are these drugs different than the ones recommended by the guideline?
	Q3	For the patient who has regimen recommendations, what are the actual dose intervals and dose durations of the administered antibiotics; and, do they follow the recommended temporal constraints?
	Q4	What are the temporal relations between administered antibiotics for that patient?
	Q5	Is there any inconsistent temporal relation between administered drugs which might occur in the antibiotic treatment ontology system?

Table 10. Clinical Questions for the Ontology

The evaluation of these rules and functions is based on the 65 medical cases listed in Appendix 1. Since each medical case has one or more corresponding reasoning rules or functions defined, the ontology is populated with relevant patient data in terms of each case to check if the relevant rules or functions can fire correctly. In order to test all of the reasoning rules and functions, a synthetic patient dataset is firstly used to test each medical case to ensure the evaluation is complete. For each one in the 65 medical cases, one or more patients are filled in the ontology. Based on the input of patient data for each medical case in the evaluation, the related reasoning rules and functions are checked if they can produce the expected output in terms of the five clinical questions. The input parameter and the expected output in the evaluation are listed in Table 11.

<table>
<tr><th>Input of Patient Information</th><th colspan="2">Expected Output</th></tr>
<tr><td rowspan="6">Clinical presentations such as diseases which are specified in the guideline (Q1)</td><td rowspan="6">Recommended antibiotic regimens</td><td>Drug name</td></tr>
<tr><td>Dose amount</td></tr>
<tr><td>Dose interval</td></tr>
<tr><td>Dose duration</td></tr>
<tr><td>Administration route</td></tr>
<tr><td>Note</td></tr>
<tr><td rowspan="2">Administered antibiotics including drug name, dose amount and administration route (Q2)</td><td colspan="2">Antibiotics which are recommended by the guideline</td></tr>
<tr><td colspan="2">Antibiotics which are not recommended by the guideline</td></tr>
<tr><td rowspan="5">Temporal information about dose periods of administered antibiotics (Q3, Q4, Q5)
• Quantitative dose periods of administered antibiotics including each dose time from start to end which are used for calculating dose interval, dose duration and inferring temporal relations
• Semi-quantitative dose periods of administered antibiotics (only start time or end time is known) which are used for inferring temporal relations
• Qualitative dose periods of administered antibiotics (both start time and end time are not known) which are used for inferring temporal relations</td><td colspan="2">Actual dose intervals of administered antibiotics and those which are not the same as the recommended intervals</td></tr>
<tr><td colspan="2">Actual dose durations of administered antibiotics and those which are not the same as the recommended durations</td></tr>
<tr><td rowspan="2">Temporal relations between administered antibiotics</td><td>Basic temporal relations</td></tr>
<tr><td>Indefinite fuzzy temporal relations</td></tr>
<tr><td colspan="2">Inconsistent or conflict temporal relations if exist in the ontology</td></tr>
</table>

Table 11. Input Parameters and Expected Outputs

In terms of input of relevant patient information, the reasoning rules and functions defined in the ontology will output the following expected results for each clinical question.

- Q1: If input the information of patient clinical presentations such as diseases, the ontology will output the correct antibiotic regimen recommendations including drug name, dose amount, dose interval, dose duration, administration route and other information in note.

- Q2: If input the information of antibiotics which are administered to patients, the ontology will find which administered antibiotics are recommended by the guideline and which ones are not.
- Q3: If input the relevant temporal information of dose periods of administered antibiotics, the ontology will calculate the actual dose intervals of these administered antibiotics; and, find which ones are same as the recommended dose intervals and which ones are not.
- Q3: If input the relevant temporal information of dose periods of administered antibiotics, the ontology will calculate the actual dose durations of these administered antibiotics; and, find which ones are same as the recommended dose durations and which ones are not.
- Q4: If input the temporal information of dose periods of administered antibiotics, the ontology will find all possible temporal relations between administered antibiotics including the basic relations and indefinite fuzzy relations.
- Q5: If inconsistent temporal relations between administered antibiotics exist in the ontology, the ontology will detect them.

For example, the following information about the patient David Brown (Table 12) is input in the ontology.

Input of Information of Patient Medical Case
Patient: David Brown
Clinical condition: sepsis (uncertain focus) and suspected meningococcal sepsis
Administered antibiotics: benzylpenicillin1 (120 mg intravenous), benzylpenicillin12 (200 mg intravenous), flucloxacillin14 (200 mg intravenous), flucloxacillin5 (200 mg oral), gentamicin5 (500 mg intravenous)
Temporal information about administered antibiotics: benzylpenicillin1 (2010-01-09T15:00:00, 2010-01-09T19:00:00, 2010-01-09T23:00:00, 2010-01-10T03:50:00, 2010-01-10T07:00:00, 2010-01-10T11:00:00, 2010-01-10T15:00:00, 2010-01-10T19:00:00, 2010-01-10T23:00:00), benzylpenicillin12 (start time:2010-01-11T03:00:00, the rest temporal information is not known), flucloxacillin5 (2010-01-09T15:00:00, 2010-01-09T21:00:00, 2010-01-10T06:00:00, 2010-01-10T12:00:00, 2010-01-10T18:00:00, 2010-01-11T00:00:00), gentamicin5 (2010-01-09T15:00:00, 2010-01-11T03:00:00), flucloxacillin14 (end time:2010-01-12T08:00:00, and it is administered after flucloxacillin5, but the rest temporal information is not known)

Table 12. An example of Patient Information

Based on the reasoning rules and functions, the ontology correctly output the following results summarised in Table 13 in terms of the five clinical questions. Result 1 (Q1) is the summary of recommended regimens for the patient with regard to his clinical conditions. Result 2 and result 3 (Q2) are about the administered antibiotics including the ones which are recommended by the guideline and the ones which are not. Since there is no administered antibiotic which is not recommended by the guideline for the patient, the output is empty in result 3. Result 4 and result 5 (Q3) are the actual dose intervals and dose durations of administered antibiotics. Result 6 (Q3) is about dose interval compliance checking. It lists all dose intervals which are not the same as the interval recommended by the guideline. Result 7 (Q3) is about the dose duration compliance checking. Since there is no dose duration requirement recommended by the guideline for this medical case, the output is empty. Result 8 and result 9 (Q4) are the inferred temporal relations between administered antibiotics for the patients. Result 10 (Q5) is the result of temporal relation inconsistency checking between patients' administered antibiotics. Since there is no inconsistent relation found, the output is empty.

Result 1: Recommended Antibiotic Regimens (Q1)						
Patient Name	**Recommended Regimen**	**Dose Agent**	**Dose Amount**	**Dose Interval**	**Administration Route**	**Note**
David Brown	Medication 1 (Sepsis, Uncertain Focus)	Flucloxa-cillin	200 mg	6 hours	Intravenous	In patients with hypersensitivities, see Antibiotic Guidelines 14
David Brown	Medication 2 (Sepsis, Uncertain Focus)	Gentamicin	7 mg/kg for 1 dose	Determine dosing interval for a maximum of either 1 or 2 further doses based on renal function (see dose interval in initial	null	In patients with hypersensitivities, see Antibiotic Guidelines 14

				Aminoglycoside (gentamicin/ tobramycin) dose)		
David Brown	Medication 4 (Sepsis, Uncertain Focus)	Benzyl-penicillin	180 mg	4 hours	Intravenous	In patients with hypersensitivities, see Antibiotic Guidelines 14

Result 2: Administered antibiotic recommended by the guideline (Q2)

Patient Name	Administered Drug	Dose Amount	Route of Administration
David Brown	benzylpenicillin1	120 mg	intravenous
David Brown	benzylpenicillin12	200 mg	intravenous
David Brown	flucloxacillin14	200 mg	intravenous
David Brown	flucloxacillin5	200 mg	oral
David Brown	gentamicin5	500 mg	intravenous

Result 3: Administered antibiotic not recommended by the guideline (Q2)

Nil

Result 4: Dose intervals of each administered antibiotic (Q3)

Patient Name	Administered Drug	Interval Start	Interval End	Length (Hours)
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-09T19:00:00	4
David Brown	benzylpenicillin1	2010-01-09T19:00:00	2010-01-09T23:00:00	4
David Brown	benzylpenicillin1	2010-01-09T23:00:00	2010-01-10T03:50:00	4.8
David Brown	benzylpenicillin1	2010-01-10T03:50:00	2010-01-10T07:00:00	3.1
David Brown	benzylpenicillin1	2010-01-10T07:00:00	2010-01-10T11:00:00	4
David Brown	benzylpenicillin1	2010-01-10T11:00:00	2010-01-10T15:00:00	4
David Brown	benzylpenicillin1	2010-01-10T15:00:00	2010-01-10T19:00:00	4
David Brown	benzylpenicillin1	2010-01-10T19:00:00	2010-01-10T23:00:00	4
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-09T21:00:00	6
David Brown	flucloxacillin5	2010-01-09T21:00:00	2010-01-10T06:00:00	9
David Brown	flucloxacillin5	2010-01-	2010-01-	6

		10T06:00:00	10T12:00:00	
David Brown	flucloxacillin5	2010-01-10T12:00:00	2010-01-10T18:00:00	6
David Brown	flucloxacillin5	2010-01-10T18:00:00	2010-01-11T00:00:00	6
David Brown	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00	36

Result 5: Dose durations of each administered antibiotic (Q3)

Patient Name	Administered Drug	Start Time	Finish Time	Duration Length (Days)
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	1.3
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	1.3
David Brown	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00	1.5

Result 6: Dose interval compliance checking (Q3)

Dose interval of administered flucloxacillin which is not equal to the recommended 6 hours interval

Patient Name	Administered Drug	Dose Time (Interval Start)	Dose Time (Interval End)	Interval Length (Hours)
David Brown	flucloxacillin5	2010-01-09T21:00:00	2010-01-10T06:00:00	9

Result 7: Dose duration compliance checking (Q3)

Not available in this category

Result 8: Basic temporal relations among administered antibiotics (Q4)

Patient Name	Administered Drug	Start Time	Finish Time	Temporal Relation	Administered Drug	Start Time	Finish Time
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	before	benzylpenicillin12	2010-01-11T03:00:00	
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	before	flucloxacillin14		2010-01-12T08:00:00
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	start	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00
David Brown	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00	start	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
David Brown	benzylpenicillin12	2010-01-11T03:00:00		after	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00

David Brown	benzylpenicillin12	2010-01-11T03:00:00		after	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00
David Brown	benzylpenicillin12	2010-01-11T03:00:00		met by	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
David Brown	flucloxacillin14		2010-01-12T08:00:00	after	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00
David Brown	flucloxacillin14		2010-01-12T08:00:00	after	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	started by	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	before	benzylpenicillin12	2010-01-11T03:00:00	
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	before	flucloxacillin14		2010-01-12T08:00:00
David Brown	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00	start	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
David Brown	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00	started by	benzylpenicillin1	2010-01-09T15:00:00	2010-01-10T23:00:00
David Brown	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00	met	benzylpenicillin12	2010-01-11T03:00:00	
David Brown	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00	started by	flucloxacillin5	2010-01-09T15:00:00	2010-01-11T00:00:00

Result 9: Fuzzy temporal relations among administered antibiotics (Q4)

Patient Name	Administered Drug	Start Time	Finish Time	Temporal Relation	Administered Drug	Start Time	Finish Time
David Brown	benzylpenicillin12	2010-01-11T03:00:00		full	flucloxacillin14		2010-01-12T08:00:00

David Brown	flucloxacillin14		2010-01-12T08:00:00	full	benzylpenicillin12	2010-01-11T03:00:00	
David Brown	flucloxacillin14		2010-01-12T08:00:00	bi_mi_oi	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00
David Brown	gentamicin5	2010-01-09T15:00:00	2010-01-11T03:00:00	b_m_o	flucloxacillin14		2010-01-12T08:00:00
Result 10: Inconsistent temporal relations among Administered Antibiotics (Q5)							
Nil							

Table 13. Produced Results from the Input of Patient Information in Table 12

Each reasoning rule and function must be defined correctly in the ontology in order to return the correct results of each medical case to clinicians. In order to ensure the correctness of the reasoning rules and functions, the following evaluation matrix (Table 14) based on the five clinical questions is developed to validate the rules and functions in each medical case. The correctness of the rules and functions is validated against the output in terms of each valuation item with the help of an ICU medical expert in our research group.

With regard to each item in the evaluation matrix, there are three types of evaluation result which are “Yes”, ”No” or ”Unavailable” to validate the rules and functions. The “Yes” answer indicates the relevant reasoning rules or functions are correct and can output the correct results whereas the “No” answer indicates the rules or functions are not correct. However, only the medical case 3 in the “Suspected Community Acquired Meningitis” category and the medical cases 1-13 in the “Trauma” category have dose duration requirement recommended by the guideline (see Appendix 1) while others do not. Therefore, the “Yes” or “No” answer to the evaluation item E7 about dose duration in the matrix is only for those cases in the “Suspected Community Acquired Meningitis” category and the “Trauma” category. For the rest cases, it is not available to answer “Yes” or “No”. Similarly, the medical case 1 in “Hospital Acquired Pneumonia” has no specific antibiotic regimen recommended by the guideline. Therefore, it is not available to answer “Yes” or “No” to the evaluation items E2, E3, E6 and E7 in this case.

Evaluation Item	Q1	E1: Can the correct regimen recommendations be found for the patients in this medical case?
	Q2	E2: Can the administered antibiotics recommended by the guideline be correctly found if there are some for the patients in this medical case?
		E3: Can the administered antibiotics not recommended by the guideline be correctly found if there are some for the patients in this medical case?
	Q3	E4: Can the dose intervals of administered antibiotics be correctly calculated with regard to the patients in this medical case?
		E5: Can the dose durations of administered antibiotics be correctly calculated with regard to the patients in this medical case?
		E6: Can the dose intervals of administered antibiotics which are not the same as the recommended one in the guideline be correctly found if there are some for the patients in this medical case?
		E7: Can the dose durations of administered antibiotics which are not the same as the recommended one in the guideline be correctly found if there are some for the patients in this medical case?
	Q4	E8: Can the basic temporal relations between administered antibiotics for the patients in this medical case be correctly found if there are some for the patients in this medical case?
		E9: Can the fuzzy temporal relations between administered antibiotics for the patients in this medical case be correctly found if there are some for the patients in this medical case?
	Q5	E10: Can the inconsistent temporal relations between the administered antibiotics for the patients in this medical case be correctly found if they exist in the ontology?
Evaluation Result for Each Item	Y	Yes
	N	No
	N/A	Unavailable to answer “Yes” or “No” in this medical case

Table 14. Evaluation Matrix

Overall 78 different patients with relevant information are filled in the ontology for the evaluation. All medical cases are covered to ensure the evaluation is complete. All the reasoning rules and functions are validated in terms of the evaluation matrix. The evaluation results are also summarised in Table 15.

Medical Case No.		Number of Tested Patients	Evaluation Results with regard to the Evaluation Matrix									
			E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
Sepsis (uncertain focus)	**1**	4 (Classie Murakami, David Brown, John Smith, Lucy Bake, Michael Jones)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	1 (Lucy Bake)	Same as above									
	3	1 (David Brown)	Same as above									
Febrile Neutropaenia	**1**	5 (Elnora Dock, Ginger Noggle, Ocie Rahm, Sherly Hickson, Yun Dobbin)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	4 (Coy Weston, Irvin Grimmer, Margot Potts, Tora Maring)	Same as above									
	3	2 (Ginger Noggle, Ocie Rahm)	Same as above									
	4	1 (Irvin Grimmer)	Same as above									
	5	1 (Yun Dobbin)	Same as above									
	6	1 (Coy Weston)	Same as above									
	7	1 (Sherly Hickson)	Same as above									
	8	1 (Tora Maring)	Same as above									
Suspected Fungal Sepsis	**1**	1 (Enid Hammon)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	3 (Bailey Stroupe, Cinthia Angert, Warner Thierry)	Same as above									

	3	1 (Bailey Stroupe)	Same as above									
	4	1 (Warner Thierry)	Same as above									
Community Acquired Pneumonia	**1**	2 (Avelina Hair, Eugene Degraw)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	2 (Beaulah Hund, Tory Ackermann)	Same as above									
	3	1 (Avelina Hair)	Same as above									
	4	1 (Beaulah Hund)	Same as above									
	5	1 (Karleen Cutrer)	Same as above									
	6	1 (Chuck Whaley)	Same as above									
Aspiration Pneumonia	**1**	4 (Aileen Ashmore, Bell Letchworth, Hettie Flatley, Kevin Majewski)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	3 (Edison Rath, Kent Reynaga, Patrick Strzelecki)	Same as above									
	3	2 (Bell Letchworth, Kevin Majewski)	Same as above									
	4	1 (Aileen Ashmore)	Same as above									
	5	1 (Kent Reynaga)	Same as above									
	6	1 (Patrick Strzelecki)	Same as above									
	7	1 (Mose Smail)	Same as above									
	8	1 (Abram Daniele)	Same as above									
	9	1 (Matt Helman)	Same as above									
	10	1 (Jona	Same as above									

		Lippincott)											
Suspected Community Acquired Meningitis	**1**	4 (Henry Jaime, Salley Buchmann, Tyson Osbourn, William Bostwick)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	
	2	1 (Henry Jaime)	Same as above										
	3	1 (Salley Buchmann)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
Trauma	**1**	1 (Iliana Felice)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	
	2	1 (Marie Jaqua)	Same as above										
	3	1 (Vito Adams)	Same as above										
	4	1 (Gregg Romans)	Same as above										
	5	1 (Rosie Quesenberry)	Same as above										
	6	1 (Cyrus Olive)	Same as above										
	7	1 (Wyatt Colbert)	Same as above										
	8	1 (Felipe Bryer)	Same as above										
	9	1 (Sonja Valenta)	Same as above										
	10	1 (Hertha Watwood)	Same as above										
	11	1 (Jefferson Chavez)	Same as above										
	12	1 (Kacey Cortinas)	Same as above										
	13	1 (Noe Lydon)	Same as above										
	14	12 (all above except Noe Lydon)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y	
Urosepsis	**1**	1 (Mathew Gramlich)	Same as above										
	2	1 (Rudy Mccarron)	Same as above										
Hospital Acquired Pneumonia	**1**	2 (Dakota Ferrer, Natisha	Y	N/A	N/A	Y	Y	N/A	N/A	Y	Y	Y	

		Hazell)										
Early Ventilator Associated Pneumonia (VAP)(provided no known colonisation with MDRO)	**1**	1 (Rebecka Janousek)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	1 (Vergie Hudock)	Same as above									
Late VAP	**1**	4 (Alexis Chickering, Margrett Woodmansee, Rob Gaulding, Shanna Heard)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	1 (Rob Gaulding)	Same as above									
	3	1 (Alexis Chickering)	Same as above									
	4	1 (Shanna Heard)	Same as above									
	5	1 (Steve Appelbaum)	Same as above									
Intra-abdominal Sepsis	**1**	1 (Valencia Gutshall)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	1 (Richard Cather)	Same as above									
Biliary Sepsis (Cholecystitis)	**1**	2 (Del Hans, Victoria Slemp)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
	2	2 (Cornell Witkowski, Rudolph Lindner)	Same as above									
	3	1 (Del Hans)	Same as above									
	4	1 (Cornell Witkowski)	Same as above									
Acute Pancreatitis	**1**	2 (Lawrence Gift, Samuel Mancini)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Total of Unique Patients Evaluated												**78**

Table 15. Summary of Tested Synthetic Patients and the Evaluation Results in Each Medical Case

As can be seen in Table 15, some patients belong to more than one medical case in a same clinical presentation category. The reason is that some medical cases are more general than other medical cases in terms of clinical conditions of patients. If a patient belongs to a more specific medical case, he or she also belongs to a more general medical case. For example, in the “Sepsis (uncertain focus)” category, if Lucy Bake is a patient who has sepsis and shock, she also is a patient who has sepsis. The inverse does not hold. The latter case is more general than the former one and is entailed by the former one. The evaluation results in Table 15 show that most of the results are marked “Y” that denotes the reasoning rules and functions in the ontology can give the correct answer in terms of the evaluation matrix. Some of them are marked “N/A” that denotes it is not available for the rules and functions to answer the questions because some medical cases do not have dose duration requirement or recommended antibiotics provided by the guideline. Thus, it is not available to check if they are same as the recommended ones in the guideline. None of them are marked “N” that denotes the questions cannot be answered correctly. Therefore, the reasoning rules and functions are all defined correctly for each medical case in the ontology.

Finally, the ontology system is evaluated using a real patient dataset to ensure it can work properly in a real environment. The real dataset is extracted from the MMIC II database which is an open source comprehensive clinical database containing clinical data from tens of thousands of Intensive Care Unit (ICU) patients collected between 2001 and 2008 in a single tertiary teaching hospital in the United States of America. The patient name and administered time of drugs are de-identified in the database for confidentiality purposes. 23 ICU patients were found in the database which can cover 14 medical cases in different clinical presentation categories (Table 16). The evaluation process is same as the one based on the synthetic dataset and the part of outputs can be found in Appendix 17. The evaluation results in Table 16 also show that the reasoning rules and functions work properly.

Medical Case No.		Number of Tested Patients	Evaluation Results with regard to the Evaluation Matrix									
			E1	E2	E3	E4	E5	E6	E7	E8	E9	E10
Sepsis (uncertain focus)	**1**	4 (Patient33, 37,222, 425)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Febrile Neutropaenia	**1**	2 (patient513, 517)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Suspected Fungal Sepsis	**2**	1 (Patient7917)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Community Acquired Pneumonia	**1**	4 (Patient 202, 253, 368, 425)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Aspiration Pneumonia	**1**	5 (Patient 9, 202,208, 222, 339)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Suspected Community Acquired Meningitis	**1**	1 (Patient 550)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Trauma	**4**	1 (Patient 172)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
	14	2 (Patient 42, 172)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Urosepsis	**1**	2 (Patient 62, 191)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Hospital Acquired Pneumonia	**1**	1 (Patient 446)	Y	N/A	N/A	Y	Y	N/A	N/A	Y	Y	Y
Early Ventilator Associated Pneumonia (VAP)	**1**	1 (Patient 897)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Late VAP	**1**	1 (Patient 405)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Intra-abdominal Sepsis	**1**	1 (Patient 946)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Biliary Sepsis (Cholecystitis)	**1**	1 (Patient 989)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Acute Pancreatitis	**1**	1 (Patient 339)	Y	Y	Y	Y	Y	Y	N/A	Y	Y	Y
Total of Patients Evaluated												**23**

Table 16. Summary of Tested Real Patients and the Evaluation Results

In summary, the evaluation of the logical consistency and clinical question answering of the extended 4D fluent approach on the QUAIC antibiotic treatment guideline ontology is analysed based on both synthetic and real patient dataset in this chapter. The evaluation result shows the ontology is logical consistent and all the reasoning rules and functions can give correct answers in terms of the five clinical questions.

Chapter 7 Conclusions

7.1 Summary of Research Contributions

In this thesis, the issue of temporal knowledge representation and reasoning in the OWL ontology-based clinical guideline systems is analysed. Due to the representation limitation of binary predicate language construct of OWL, the traditional OWL-based clinical guideline systems do not support temporal knowledge representation and reasoning. This limitation prevents a wider application of OWL, e.g., clinical guideline systems. In this thesis, an extended 4D fluent temporal knowledge representation method is presented to deal with the shortcoming of the traditional OWL-based clinical guideline systems. By leveraging the extended 4D fluent method, it is possible to model valid calendar time, repetitive temporal constraints and temporal relations, and implement the related temporal knowledge reasoning in the OWL-based clinical guideline systems. The extended 4D fluent method is demonstrated in a prototype of OWL-based antibiotic treatment guideline ontology system which is derived from the QUAIC guidelines. In the prototype guideline ontology system, clinical knowledge and temporal knowledge contained in the antibiotic regimen recommendations of the guidelines are represented in a domain ontology and an extended 4D fluent ontology respectively. Moreover, Oracle user defined reasoning rules and functions are leveraged to develop a knowledge reasoning system which can assist clinicians to research and review their antibiotic administration practice with regard to the guidelines. The analysis in previous chapters shows that the reasoning system can answer the clinical questions about antibiotic regimen recommendations and the temporal-related questions which rely on the extended 4D fluent method such as administered antibiotics, dose intervals of administered antibiotics, dose durations of administered antibiotics, exact temporal relations between administered antibiotics and inconsistent temporal relations between administered antibiotics. Therefore, the shortcoming of temporal knowledge representation and reasoning in the traditional OWL-based clinical guideline systems has been overcome to an extent.

7.2 Limitations of the Research and Future Work

Although the original 4D fluent temporal knowledge modelling method is extended to enable the temporal knowledge representation and reasoning in the OWL-based clinical guideline systems, the types of temporal constraint in clinical activities or events which are modelled in the extended 4D fluent method are still limited, i.e., it only handles the valid calendar time, the repetitive temporal constraint and temporal relations. Although the repetitive temporal constraint and the temporal relation constraint are important to clinical tasks such as antibiotic administration, some other important clinical tasks such as clinical activity scheduling in various clinical guidelines not only involve the repetitive temporal constraint and the temporal relation constraint but also involve constraints about temporal relativity, indeterminacy or uncertainty, and delay. In order to deal with the clinical scheduling tasks specified in different guidelines, the extended 4D fluent method needs to be further extended to model all the related temporal constraints.

Another major limitation of this research is that knowledge acquisition in the OWL-based clinical guideline systems is more difficult under the extended 4D fluent method. It is well known that the manual construction of ontologies is a time consuming task due to the complex OWL logical syntax. Like other temporal knowledge representation methods such as RDF reification and N-ary relation reification methods, the extended 4D fluent method also requires rewriting of the source ontologies to include extra classes, instances and relations for modelling the relations which hold in time in the domain of discourse. Thus, it will add more statements to the original ontology and lead to the object proliferation issue in the ontology. Moreover, it is more difficult to populate the ontologies due to the complexity of the modelling method. In addition, the complexity of the populated ontology will bring extra difficulties to the maintenance of the ontology.

With regard to these limitations in this research, the future work will focus on two aspects. One aspect is to investigate the approaches which can extend the 4D fluent method to model the temporal constraints for scheduling tasks in OWL-based clinical guideline systems. Another aspect is to investigate software tools which can mitigate the burden of knowledge acquisition under the extended 4D fluent modelling method.

Appendices

Appendix 1-Patient Medical Case Classification and Recommended Regimen

Patient Clinical Conditions			Antibiotic Regimen
Community Presentation	Sepsis (uncertain focus)	1. Patient who has sepsis	Flucloxacillin (200 mg IV 6 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		2. Patient who has sepsis and shock	Flucloxacillin (200 mg IV 6 hourly); Vancomycin (150 mg IV 12 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		3. Patient who has sepsis and suspected meningococcal sepsis	Flucloxacillin (200 mg IV 6 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function); Benzylpenicillin (180 mg IV 4 hourly)
		4. Patient who has sepsis and hypersensitivities	No medical recommendation available; See Antibiotic Guidelines 14
	Febrile Neutropaenia	1. Patient who has febrile neutropenia, but has not had penicillin hypersensitivity	Piperacillin/Tazobactam (400 mg + 50 mg IV 8 hourly)
		2. Patient who has febrile neutropenia patient and minor penicillin hypersensitivity	Ceftazidime (200 mg IV 8 hourly)
		3. Patient who has	Piperacillin/Tazobactam (400

		febrile neutropenia and shock, but has not had penicillin hypersensitivity	mg + 50 mg IV 8 hourly); Vancomycin (150 mg IV 12 hourly)
		4. Patient who has febrile neutropenia, shock and minor penicillin hypersensitivity	Vancomycin (150 mg IV 12 hourly); Ceftazidime (200 mg IV 8 hourly)
		5. Patient who has febrile neutropenia Patient and methicillin-resistant staphylococcus aureus (MRSA) colonisation, but has not had penicillin hypersensitivity	Piperacillin/Tazobactam (400 mg + 50 mg IV 8 hourly); Vancomycin (150 mg IV 12 hourly)
		6. Patient who has febrile neutropenia patient, MRSA colonisation and minor penicillin hypersensitivity	Vancomycin (150 mg IV 12 hourly); Ceftazidime (200 mg IV 8 hourly)
		7. Patient who has febrile neutropenia, catheter related infection in a unit with a high incidence of MRSA infection, but has not had penicillin hypersensitivity	Piperacillin/Tazobactam (400 mg + 50 mg IV 8 hourly); Vancomycin (150 mg IV 12 hourly
		8. Patient who has febrile neutropenia, catheter related infection in a unit with a high incidence of MRSA infection and minor penicillin hypersensitivity	Vancomycin (150 mg IV 12 hourly); Ceftazidime (200 mg IV 8 hourly)
	Suspected Fungal Sepsis	1. Patient who has suspected fungal sepsis and is azole naïve, but has not had candida	Fluconazole (800 mg first dose, 400 mg IV 24 hourly)

		glabrata isolation and candida kruzei isolation	
		2. Patient who has suspected fungal sepsis but is not azole naïve	Amphotericin B (0.5 to 1 mg/kg IV 24 hourly) **OR** Caspofungin (70mg IV first dose, then 50mg IV 24 hourly)
		3. Patient who has suspected fungal sepsis and candida glabrata isolation	Amphotericin B (0.5 to 1 mg/kg IV 24 hourly) **OR** Caspofungin (70mg IV first dose, then 50mg IV 24 hourly)
		4. Patient who has suspected fungal sepsis and candida kruzei isolation	Amphotericin B (0.5 to 1 mg/kg IV 24 hourly) **OR** Caspofungin (70mg IV first dose, then 50mg IV 24 hourly)
	Community Acquired Pneumonia	7. Patient who has community acquired pneumonia, but has not had severe sepsis and penicillin hypersensitivity	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly)
		8. Patient who has community acquired pneumonia and severe sepsis, but has not had penicillin hypersensitivity	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly)
		9. Patient who has community acquired pneumonia and suspected staphylococcal pneumonia, but has not had penicillin hypersensitivity and severe sepsis	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly);

			Vancomycin (150 mg IV 12 hourly)
		10. Patient who has community acquired pneumonia, suspected staphylococcal pneumonia and severe sepsis, but has not had penicillin hypersensitivity	Benzylpenicillin (1.2g IV 4 hourly), Azithromycin (500mg IV 24 hourly) and Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Azithromycin (500mg IV 24 hourly) and Ceftriaxone (1g IV 24 hourly); Vancomycin (150 mg IV 12 hourly)
		11. Patient who has community acquired pneumonia and immediate penicillin hypersensitivity	Azithromycin (500 mg IV 24 hourly); Moxifloxacin (400 mg IV 24 hourly)
		12. Patient who has severe community acquired pneumonia and severe influenza that is in the period when influenza A virus is circulating	Neuramindase Inhibitor (Oseltamivir **OR** Zanamivir) (150 mg nasogastric tube 12 hourly)
	Aspiration Pneumonia	1. Patient who aspiration pneumonia, but has not had penicillin hypersensitivity and pseudomonal pneumonia	Metronidazole (500 mg IV 12 hourly); Benzylpenicillin (120 mg IV 4 hourly)
		2. Patient who has aspiration pneumonia and immediate penicillin hypersensitivity, but has not had pseudomonal pneumonia	Lincomycin (600mg IV 8 hourly) **OR** Clindamycin (450mg IV 8 hourly)
		3. Patient who has aspiration pneumonia and suspected aerobic gram negatives, but has not had severe	Metronidazole (500 mg IV 12 hourly); Benzylpenicillin (120 mg IV 4 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing

		sepsis, penicillin hypersensitivity and pseudomonal pneumonia	interval for a maximum of either 1 or 2 further doses based on renal function)
		4. Patient who has aspiration pneumonia, suspected aerobic gram negatives and severe sepsis, but has not had penicillin hypersensitivity and pseudomonal pneumonia	Metronidazole (500 mg IV 12 hourly); Benzylpenicillin (120 mg IV 4 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		5. Patient who has aspiration pneumonia, suspected aerobic gram negatives and immediate penicillin hypersensitivity, but has not had severe sepsis and pseudomonal pneumonia	Lincomycin (600mg IV 8 hourly) **OR** Clindamycin (450mg IV 8 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		6. Patient who has aspiration pneumonia, suspected aerobic Gram negatives, immediate penicillin hypersensitivity and severe sepsis, but has not had pseudomonal pneumonia	Lincomycin (600mg IV 8 hourly) **OR** Clindamycin (450mg IV 8 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		7. Patient who has aspiration pneumonia and pseudomonal pneumonia, but has not had severe sepsis and penicillin hypersensitivity	Piperacillin/Tazobactam (400 mg+50 mg IV 6 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		8. Patient who has aspiration	Piperacillin/Tazobactam (400 mg+50 mg IV 6 hourly);

		pneumonia, pseudomonal pneumonia and severe sepsis, but has not had penicillin hypersensitivity	Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		9. Patient who has aspiration pneumonia, pseudomonal pneumonia and minor penicillin hypersensitivity, but has not had severe sepsis	Ceftazidime (200 mg IV 8 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		10. Patient who has aspiration pneumonia, pseudomonal pneumonia, minor penicillin hypersensitivity and severe sepsis	Ceftazidime (200 mg IV 8 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
	Suspected Community Acquired Meningitis	1. Patient who has suspected community acquired meningitis	Ceftriaxone (4g IV 24 hourly) **OR** Cefotaxime (2g IV 6 hourly)
		2. Patient who has suspected community acquired meningitis and the risk of listeria infection	Ceftriaxone (4g IV 24 hourly) **OR** Cefotaxime (2g IV 6 hourly); Benzylpenicillin (240 mg IV 4 hourly)
		3. Patient who has suspected community acquired meningitis and herpes simplex encephalitis	Ceftriaxone (4g IV 24 hourly) **OR** Cefotaxime (2g IV 6 hourly); Acyclovir (10 mg/kg IV 8 hourly minimum 14 days)
	Trauma	1. Patient who has Gustillo type I (wound closed) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 1 day) **OR** Vancomycin (1.5g IV 12 hourly 1day)
		2. Patient who has Gustillo type I	Cefazolin (2g IV 8 hourly 2 days) **OR** Vancomycin (1.5g IV 12

		(wound open) non elective orthopaedic trauma	hourly 2 days)
		3. Patient who has Gustillo type II (wound closed) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 1 day) **OR** Vancomycin (1.5g IV 12 hourly 1 day)
		4. Patient who has Gustillo type II (wound open) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 3 days) **OR** Vancomycin (1.5g IV 12 hourly 3 days)
		5. Patient who has Gustillo type III (wound closed) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 1 day) **OR** Vancomycin (1.5g IV 12 hourly 1 day)
		6. Patient who has Gustillo type III (wound open) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 5 days) **OR** Vancomycin (1.5g IV 12 hourly 5 days)
		7. Patient who has Gustillo type IIIA (wound closed) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 1 day) **OR** Vancomycin (1.5g IV 12 hourly 1day)
		8. Patient who has Gustillo type IIIA (wound open) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 5 days) **OR** Vancomycin (1.5g IV 12 hourly 5 days)
		9. Patient who has Gustillo type IIIB (wound closed) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 1 day) **OR** Vancomycin (1.5g IV 12 hourly 1day)
		10. Patient who has Gustillo type IIIB (wound open) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 5 days) **OR** Vancomycin (1.5g IV 12 hourly 5 days)
		11. Patient who has Gustillo type IIIC (wound closed) non elective orthopaedic	Cefazolin (2g IV 8 hourly 1 day) **OR** Vancomycin (1.5g IV 12 hourly 1 day)

		trauma	
		12. Patient who has Gustillo type IIIC (wound open) non elective orthopaedic trauma	Cefazolin (2g IV 8 hourly 5 days) **OR** Vancomycin (1.5g IV 12 hourly 5 days)
		13. Patient who has other multi–trauma including brain injury, base of skull fracture and CSF monitoring in place	Cefazolin (2g IV 8 hourly 1 day)
		14. Patient who has non elective orthopaedic trauma without knowing the Gustillo type (general case)	Cefazolin (2g IV 8 hourly) **OR** Vancomycin (1.5g IV 12 hourly)
	Urosepsis	1. Patient who has urosepsis, but has not had severe sepsis	Ampicillin (200 mg IV 6 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		2. Patient who has urosepsis and severe sepsis	Ampicillin (200 mg IV 6 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
Health Care Associated Presentation (high risk of MDRO or known MDRO colonisation)	Hospital Acquired Pneumonia (HAP)	1. Patient who has hospital acquired pneumonia	Only general medical recommendation available
	Early Ventilator Associated Pneumonia (VAP)(provided no known colonisation with MDRO)	1. Patient who has early VAP, but has not had severe sepsis	Benzylpenicillin (1.2g IV 6 hourly) + Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) **OR** Ceftriaxone (1g IV 24 hourly)
		2. Patient who has early VAP and severe sepsis	Benzylpenicillin (1.2g IV 6 hourly) + Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on

			renal function) **OR** Ceftriaxone (1g IV 24 hourly)
	Late VAP	1. Patient who has late VAP, but has not had penicillin hypersensitivity	Piperacillin/Tazobactam (400 mg + 50 mg IV 6 hourly)
		2. Patient who has late VAP and is ventilated, but has not had penicillin hypersensitivity and severe sepsis	Piperacillin/Tazobactam (400 mg + 50 mg IV 6 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		3. Patient who has late VAP, severe sepsis and is ventilated, but has not had penicillin hypersensitivity	Piperacillin/Tazobactam (400 mg + 50 mg IV 6 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function; add it within maximum 48 hours)
		4. Patient who has late VAP and MRSA colonization, but has not had penicillin hypersensitivity	Piperacillin/Tazobactam (400 mg + 50 mg IV 6 hourly); Vancomycin (150 mg IV 12 hourly)
		5. Patient who has late VAP and minor penicillin hypersensitivity	Cefepime (200 mg IV 8 hourly)
	Intra-abdominal Sepsis	1. Patient who has intra-abdominal sepsis, but has not had severe sepsis	Ampicillin (1g IV 6hourly); Metronidazole (500mg IV 12 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) Change to Piperacillin/Tazobactam (4+0.5g IV 6 hourly) **OR** Ticarcillin/Clavulanate (3+0.1g IV 6 hourly) if patient is still septic

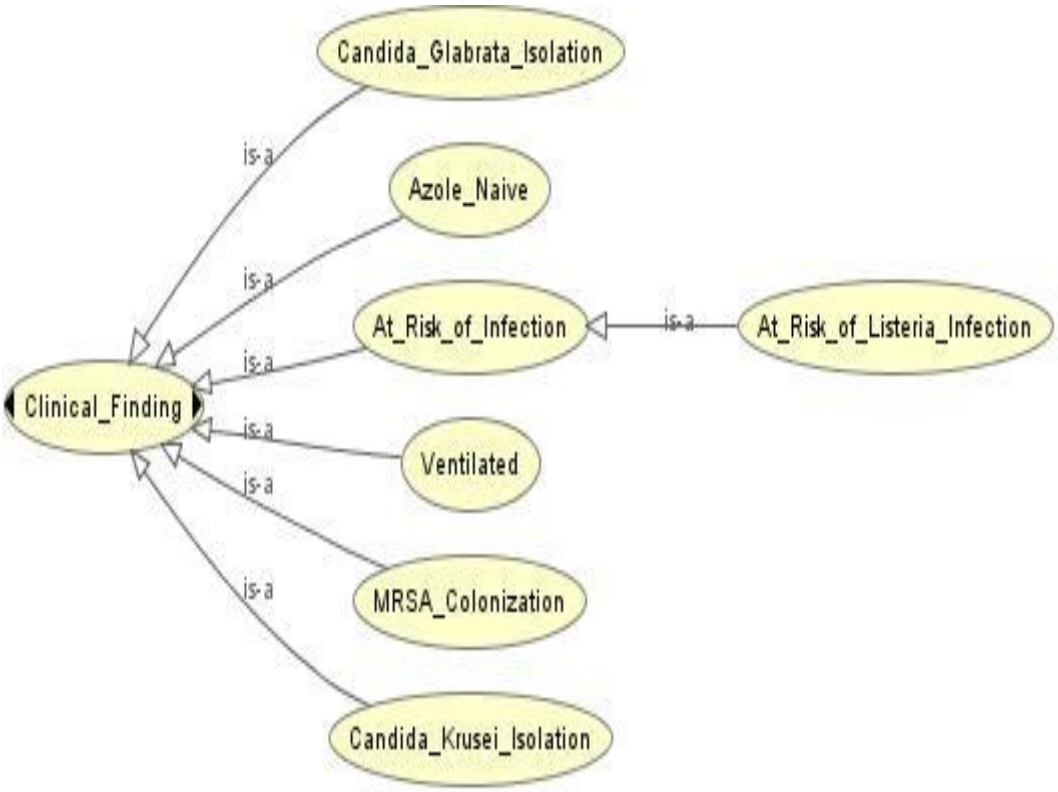
		2. Patient who has intra-abdominal sepsis and severe sepsis	Ampicillin (1g IV 6hourly); Metronidazole (500mg IV 12 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function) Change to Piperacillin/Tazobactam (4+0.5g IV 6 hourly) **OR** Ticarcillin/Clavulanate (3+0.1g IV 6 hourly) if patient is still septic
	Biliary Sepsis (Cholecystitis)	1. Patient who has biliary sepsis, but has not had severe sepsis	Ampicillin (100 mg IV 6 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		2. Patient who has biliary sepsis and severe sepsis	Ampicillin (100 mg IV 6 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function)
		3. Patient who has biliary sepsis and biliary obstruction, but has not had severe sepsis	Ampicillin (100 mg IV 6 hourly); Gentamicin (4-6 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function); Metronidazole (500 mg IV 12 hourly)
		4. Patient who has biliary sepsis, biliary obstruction and severe sepsis	Ampicillin (100 mg IV 6 hourly); Gentamicin (7 mg/kg for 1dose, determine dosing interval for a maximum of either 1 or 2 further doses based on renal function); Metronidazole (500 mg IV 12 hourly)
	Acute Pancreatitis	1. Patient who has acute pancreatitis	Piperacillin/Tazobactam (400 mg + 50 mg IV 8 hourly)

Appendix 2-Clinical Finding Class Hierarchy

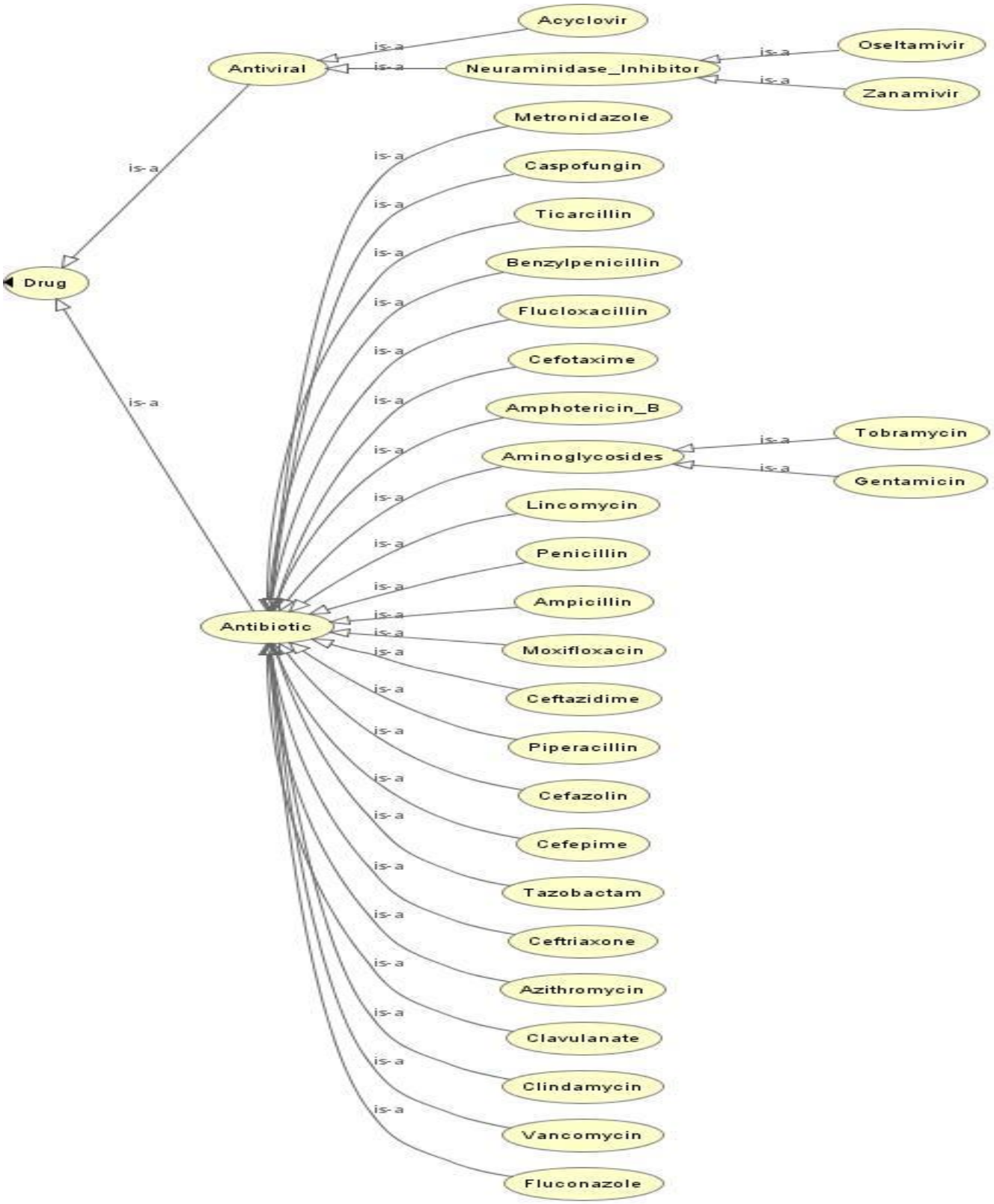
Part 1-Disease Class Hierarchy


Clinical_Finding
is-a
Disease
is-a
Endocarditis
is-a
Streptococcal_Endocarditis
is-a
Enterococcal_Endocarditis
is-a
Trauma
is-a
Other_Multi-Trauma
is-a
Non_Elective_Orthopaedic_Trauma
is-a
Non_Elective_Orthopaedic_Trauma_GTypeII_with_Wound_Closed
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIIIA_with_Wound_Open
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIIIC_with_Wound_Closed
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIII_with_Wound_Closed
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIIIA_with_Wound_Closed
is-a
Non_Elective_Orthopaedic_Trauma_GTypeI_with_Wound_Closed
is-a
Non_Elective_Orthopaedic_Trauma_GTypeII_with_Wound_Open
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIIIB_with_Wound_Closed
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIIIC_with_Wound_Open
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIII_with_Wound_Open
is-a
Non_Elective_Orthopaedic_Trauma_GTypeIIIB_with_Wound_Open
is-a
Non_Elective_Orthopaedic_Trauma_GTypeI_with_Wound_Open
is-a
Multiple-Resistant_Staphylococcus_Aureus_Infection
is-a
Hypersensitivity
is-a
Penicillin_Hypersensitivity
is-a
Immediate_Penicillin_Hypersensitivity
is-a
Minor_Penicillin_Hypersensitivity
is-a
Infection
is-a
Catheter_Related_Infection
is-a
Catheter_Related_Infection_in_a_Unit_with_a_High_Incidence_of_MRSA_Infection
is-a
Influenza
is-a
Severe_Influenza_When_InfluenzaA_Circulating
is-a
Aerobic_Gram_Negative
is-a
Suspected_Aerobic_Gram_Negative

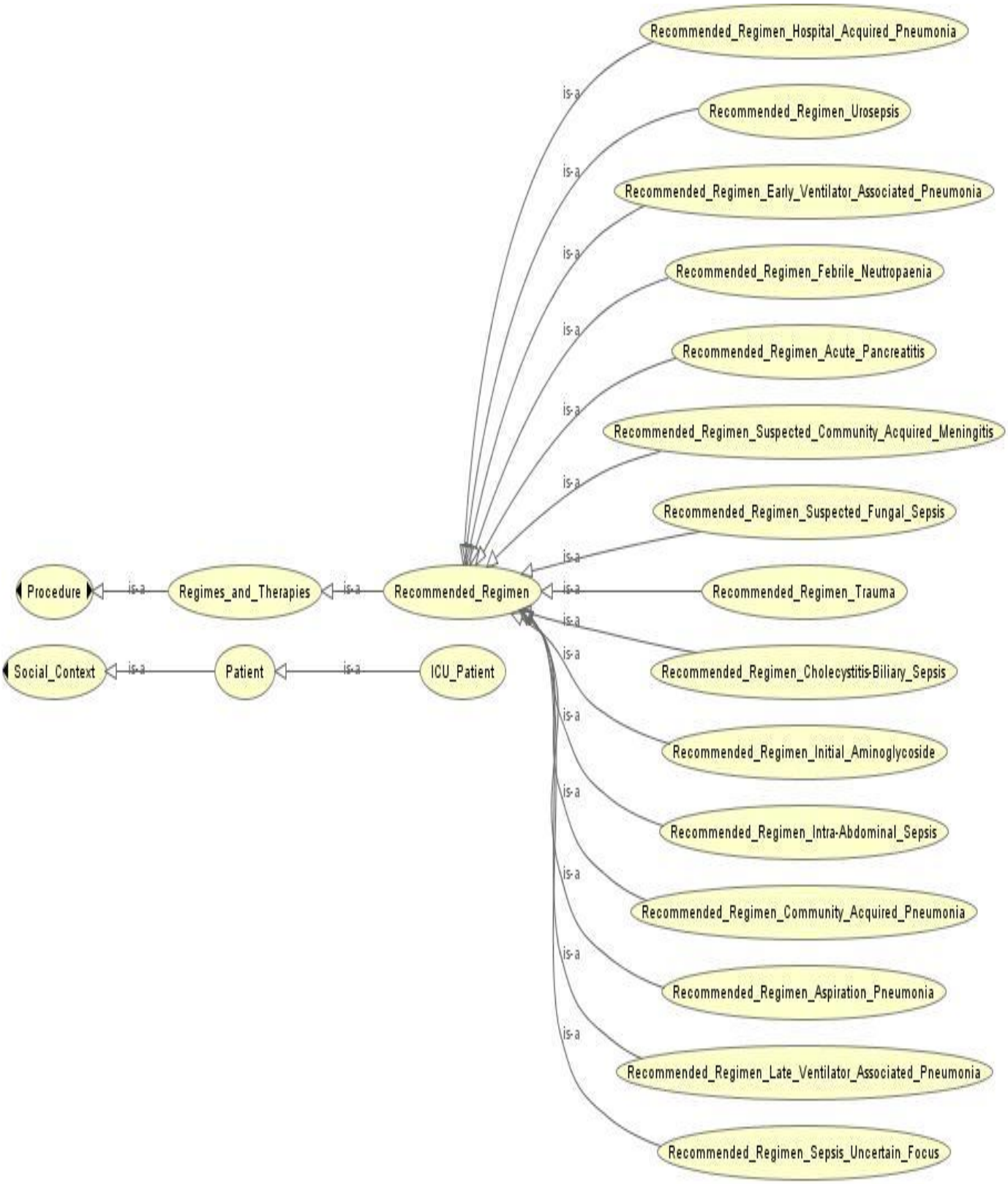
Part 2-Other Clinical Presentation Class Hierarchy



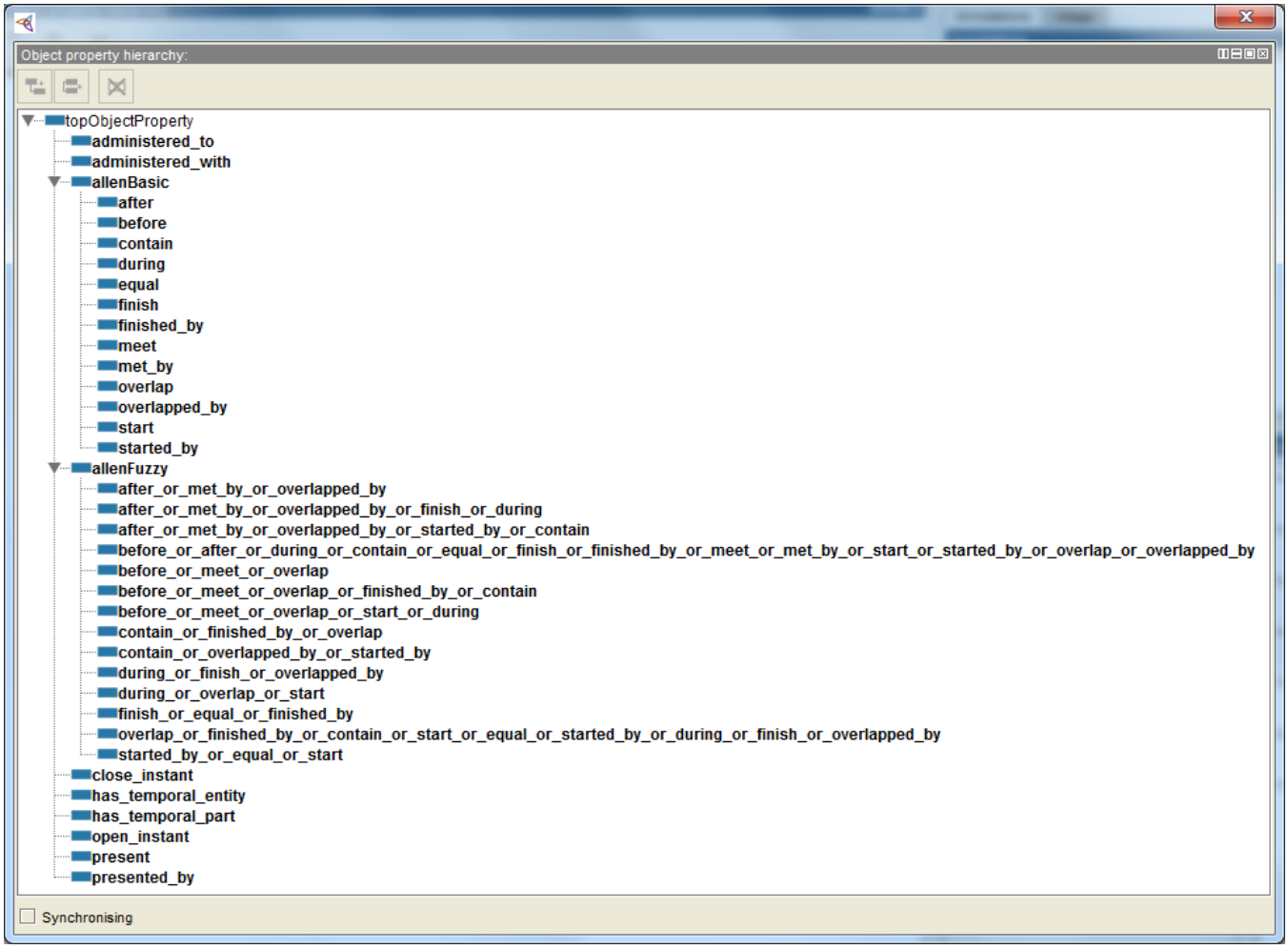
Appendix 3-Drug Class Hierarchy



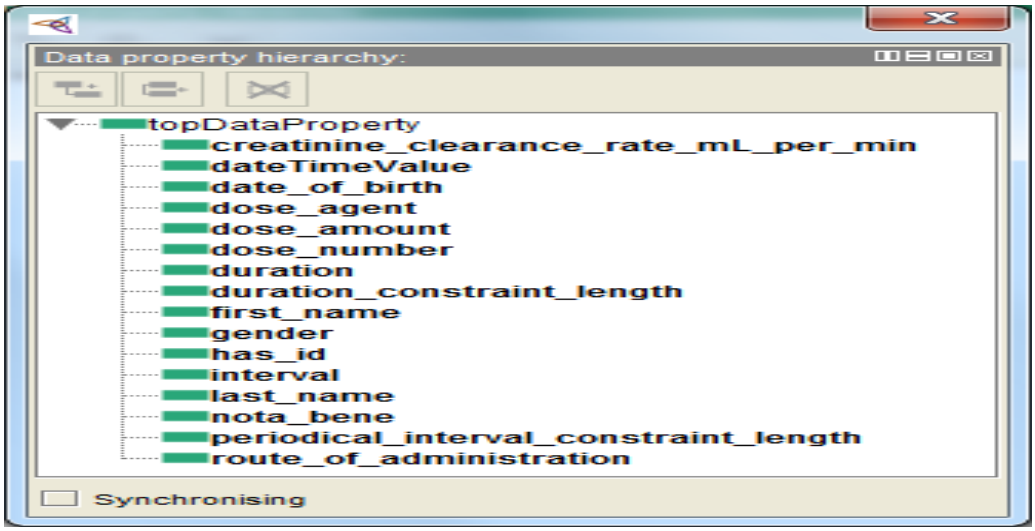
Appendix 4-Procedurce and Social Context Class Hierarchy



Appendix 5-Object Property and its Inverse Property



Appendix 6- Data Property



Appendix 7-The Extended 4D Fluent Ontology

Appendix 8 –User Defined Rules for Finding Regimen Recommendations (Part)

```
--Case 1: patient who has sepsis

INSERT INTO mdsys.semr_cas1_2 VALUES(
'rule1',
'(?p rdf:type :ICU_Patient) (?p :present ?c) (?c rdf:type :Sepsis_Uncertain_Focus)',
null,
'(?p :has_recommendation :c_sepsis_medication1)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));


INSERT INTO mdsys.semr_cas1_2 VALUES(
'rule2',
'(?p rdf:type :ICU_Patient) (?p :present ?c) (?c rdf:type :Sepsis_Uncertain_Focus)',
null,
'(?p :has_recommendation :c_sepsis_medication2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));


--Case 2: patient who has sepsis and shock

INSERT INTO mdsys.semr_cas3_4_5 VALUES(
'rule3',
'(?p rdf:type :ICU_Patient) (?p :present ?c1) (?c1 rdf:type :Sepsis_Uncertain_Focus)
 (?p :present ?c2) (?c2 rdf:type :Shock)',
null,
'(?p :has_recommendation :c_sepsis_medication1)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_cas3_4_5 VALUES(
'rule4',
'(?p rdf:type :ICU_Patient) (?p :present ?c1) (?c1 rdf:type :Sepsis_Uncertain_Focus)
 (?p :present ?c2) (?c2 rdf:type :Shock)',
null,
'(?p :has_recommendation :c_sepsis_medication2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_cas3_4_5 VALUES(
'rule5',
'(?p rdf:type :ICU_Patient) (?p :present ?c1) (?c1 rdf:type :Sepsis_Uncertain_Focus)
 (?p :present ?c2) (?c2 rdf:type :Shock)',
null,
'(?p :has_recommendation :c_sepsis_medication3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

```

```
--Case 3: patient who has sepsis and suspected meningococcal sepsis

INSERT INTO mdsys.semr_cas6_7_8 VALUES(
'rule6',
'(?p rdf:type :ICU_Patient) (?p :present ?c1) (?c1 rdf:type :Sepsis_Uncertain_Focus)
 (?p :present ?c2) (?c2 rdf:type :Suspected_Meningococcal_Sepsis)',
null,
'(?p :has_recommendation :c_sepsis_medication1)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_cas6_7_8 VALUES(
'rule7',
'(?p rdf:type :ICU_Patient) (?p :present ?c1) (?c1 rdf:type :Sepsis_Uncertain_Focus)
 (?p :present ?c2) (?c2 rdf:type :Suspected_Meningococcal_Sepsis)',
null,
'(?p :has_recommendation :c_sepsis_medication2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_cas6_7_8 VALUES(
'rule8',
'(?p rdf:type :ICU_Patient) (?p :present ?c1) (?c1 rdf:type :Sepsis_Uncertain_Focus)
 (?p :present ?c2) (?c2 rdf:type :Suspected_Meningococcal_Sepsis)',
null,
'(?p :has_recommendation :c_sepsis_medication4)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

Appendix 9-An Example of User Defined Inference Function for Negation

```
--Case 1: patient has community acquired aspiration pneumonia, but has not had penicillin hypersensitivity and pseudomonal pneumonia

create or replace function sem_inf_caap_negation_rule1 (
src_tab_view in varchar2,
resource_id_map_view in varchar2,
output_tab in varchar2,
action in varchar2,
num_calls in number,
tplInferredLastRound in number,
options in varchar2 default null,
optimization_flag out number,
diag_message out varchar2 )
return boolean
as
patientClassId number;
caapClassId number;
phClassId number;
ppClassId number;
rdfTypePropertyId number;
presentPropertyId number;
recomPropertyId number;
medId1 number;
medId2 number;
sqlStmt1 varchar2(4000);
insertStmt1 varchar2(4000);

pragma autonomous_transaction;
begin
if (action = 'RUN') then
patientClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#ICU_Patient');
caapClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Aspiration_Pneumonia');
phClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Penicillin_Hypersensitivity');
ppClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Pseudomonal_Pneumonia');
rdfTypePropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.w3.org/1999/02/22-rdf-syntax-ns#type');
presentPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#present');
recomPropertyId := sdo_sem_inference.oracle_orardf_add_res('http://www.usyd.edu.au/hitru/antibiotics#has_recommendation');
medId1 := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#caap_medication1');
medId2 := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#caap_medication2');

--negation rule1

```

```
sqlStmt1 :=
'select ids1.sid patientId
from
' || src_tab_view || ' ids1,
' || src_tab_view || ' ids2,
' || src_tab_view || ' ids3
where ids1.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND ids1.oid = ' || to_char(patientClassId,'TM9') || '

AND ids2.pid = '|| to_char(rdfTypePropertyId,'TM9') ||'
AND ids2.oid = ' || to_char(caapClassId,'TM9') || '

AND ids1.sid = ids3.sid
AND ids3.pid = ' || to_char(presentPropertyId,'TM9') || '
AND ids3.oid = ids2.sid

AND not exists
(select 1
from ' || src_tab_view || ' ids4,
     ' || src_tab_view || ' ids5
where ids4.pid = '|| to_char(rdfTypePropertyId,'TM9') || '
AND ids4.oid = ' || to_char(phClassId,'TM9') || '
AND ids1.sid = ids5.sid
AND ids5.pid = ' || to_char(presentPropertyId,'TM9') || '
AND ids5.oid = ids4.sid )

AND not exists
(select 1
from ' || src_tab_view || ' ids6,
     ' || src_tab_view || ' ids7
where ids6.pid = '|| to_char(rdfTypePropertyId,'TM9') || '
AND ids6.oid = ' || to_char(ppClassId,'TM9') || '
AND ids1.sid = ids7.sid
AND ids7.pid = ' || to_char(presentPropertyId,'TM9') || '
AND ids7.oid = ids6.sid )
';

insertStmt1 :=
'insert  into ' || output_tab || ' (sid, pid, oid)
select patientId,
' || to_char(recomPropertyId,'TM9') || ',
' || to_char(medId1, 'TM9') ||'
from (' || sqlStmt1 || ')
UNION
select patientId,
' || to_char(recomPropertyId,'TM9') || ',
' || to_char(medId2, 'TM9') ||'
from (' || sqlStmt1 || ')
';

execute immediate insertStmt1;
commit;
end if;
optimization_flag := SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_NEWDATA_ONLY +
SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_UNIQDATA_ONLY;
return true;
exception
when others then
diag_message := 'error occurred: ' || SQLERRM;
return false;
end sem_inf_caap_negation_rule1;
```

Appendix 10- Inference Function for Dose Interval Calculation

```
create or replace
function sem_inf_timeInterval(
 src_tab_view in varchar2,
 resource_id_map_view in varchar2,
 output_tab in varchar2,
 action in varchar2,
 num_calls in number,
 tplInferredLastRound in number,
 options in varchar2 default null,
 optimization_flag out number,
 diag_message out varchar2
 )
 return boolean
 as
 timeInstantClassId number;
 timeIntervalClassId number;
 rdfTypePropertyId number;
 openInstantPropertyId number;
 closeInstantPropertyId number;
 dateTimeValPropertyId number;
 timeIntervalPropertyId number;
 xsdTimeFormat varchar2(100);
 sqlStmt varchar2(4000);
 insertStmt varchar2(4000);
 pragma autonomous_transaction;
 begin
if (action = 'RUN') then
 -- retrieve ID of resource that already exists in the data (will
 -- throw exception if resource does not exist).
 timeInstantClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Time_Instant');
 timeIntervalClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Dose_Interval');
 openInstantPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#open_instant');
 closeInstantPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#close_instant');
 dateTimeValPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#dateTimeValue');
 timeIntervalPropertyId := sdo_sem_inference.oracle_orardf_add_res('http://www.usyd.edu.au/hitru/antibiotics#interval_hourly');
 rdfTypePropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.w3.org/1999/02/22-rdf-syntax-ns#type');
 -- set the TIMESTAMP format to parse XSD times
 xsdTimeFormat := 'YYYY-MM-DD"T"HH24:MI:SSTZH:TZM';
 -- query to extract the dose interval ID and its open endpoints/close endpoints, and time values of endpoints.
 sqlStmt :=
 'select ids1.sid timeIntervalInstance, ids2.sid openTimeInstant, ids3.sid closeTimeInstant,
 TO_TIMESTAMP_TZ(values1.value_name,''YYYY-MM-DD"T"HH24:MI:SSTZH:TZM'') openTime,
 TO_TIMESTAMP_TZ(values2.value_name,''YYYY-MM-DD"T"HH24:MI:SSTZH:TZM'') closeTime
 from ' || resource_id_map_view || ' values1,
 ' || resource_id_map_view || ' values2,
 ' || src_tab_view || ' ids1,
 ' || src_tab_view || ' ids2,
 ' || src_tab_view || ' ids3,
 ' || src_tab_view || ' ids4,
 ' || src_tab_view || ' ids5,
```

```
' || src_tab_view || ' ids5,
' || src_tab_view || ' ids6,
' || src_tab_view || ' ids7

where idsl.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND idsl.oid = ' || to_char(timeIntervalClassId,'TM9') || '

AND ids2.pid = '|| to_char(rdfTypePropertyId,'TM9')||'
AND ids2.oid = '|| to_char(timeInstantClassId,'TM9') ||'
AND ids3.pid ='|| to_char(rdfTypePropertyId,'TM9')||'
AND ids3.oid = '|| to_char(timeInstantClassId,'TM9') ||'

AND ids4.sid = idsl.sid
AND ids4.pid = ' || to_char(openInstantPropertyId,'TM9') || '
AND ids4.oid = ids2.sid

AND ids5.sid = idsl.sid
AND ids5.pid = ' || to_char(closeInstantPropertyId,'TM9') || '
AND ids5.oid = ids3.sid

AND ids6.sid = ids2.sid
AND ids6.pid = ' || to_char (dateTimeValPropertyId,'TM9') || '
AND ids6.oid = valuesl.value_id

AND ids7.sid = ids3.sid
AND ids7.pid = ' || to_char (dateTimeValPropertyId,'TM9') || '
AND ids7.oid = values2.value_id


AND not exists
(select 1
from ' || src_tab_view || '
where sid = idsl.sid
AND pid = ' || to_char(timeIntervalPropertyId,'TM9') || ')

AND not exists
(select 1
from ' || output_tab || '
where sid = idsl.sid
AND pid = ' || to_char(timeIntervalPropertyId,'TM9') || ')';

-- compute the difference (in hours) between the two Oracle
-- timestamps from the sqlStmt query. Store the hours as
-- xsd:decimal.
insertStmt :=
'insert  into ' || output_tab || ' (sid, pid, o)
select timeIntervalInstance,
' || to_char(timeIntervalPropertyId,'TM9') || ',
''"'' || hours || ''"^^xsd:decimal''
from (
```

```
select timeIntervalInstance,
( trunc(
      (extract(day from (closeTime - openTime))*24 +
       extract (hour from (closeTime - openTime)) +
       extract (minute from (closeTime - openTime))/60),1
      )
) hours
from (' || sqlStmt || '))';
-- execute the query
execute immediate insertStmt;
-- commit our changes
commit;
end if;

optimization_flag := SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_NEWDATA_ONLY +
SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_UNIQDATA_ONLY;
-- return true to indicate success
return true;
-- handle any exceptions
exception
when others then
diag_message := 'error occurred: ' || SQLERRM;
return false;
end sem_inf_timeInterval;
```

Appendix 11- Inference Function for Dose Duration Calculation

```
create or replace
function sem_inf_timeDuration(
src_tab_view in varchar2,
resource_id_map_view in varchar2,
output_tab in varchar2,
action in varchar2,
num_calls in number,
tplInferredLastRound in number,
options in varchar2 default null,
optimization_flag out number,
diag_message out varchar2
)
return boolean
as
timeInstantClassId number;
timePeriodClassId number;
rdfTypePropertyId number;
openInstantPropertyId number;
closeInstantPropertyId number;
dateTimeValPropertyId number;
timeDurationPropertyId number;
xsdTimeFormat varchar2(100);
sqlStmt varchar2(4000);
insertStmt varchar2(4000);
pragma autonomous_transaction;
begin
if (action = 'RUN') then
-- retrieve ID of resource that already exists in the data (will
-- throw exception if resource does not exist).
timeInstantClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Time_Instant');
timePeriodClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Quantitative_Dose_Period');
openInstantPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#open_instant');
closeInstantPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#close_instant');
dateTimeValPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#dateTimeValue');
timeDurationPropertyId := sdo_sem_inference.oracle_orardf_add_res('http://www.usyd.edu.au/hitru/antibiotics#duration_days');
rdfTypePropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.w3.org/1999/02/22-rdf-syntax-ns#type');
-- set the TIMESTAMP format to parse XSD times
xsdTimeFormat := 'YYYY-MM-DD"T"HH24:MI:SSTZH:TZM';
-- sql query to extract the dose period ID, its start/end times and their values.
sqlStmt :=
'select ids1.sid timePeriodInstance, ids2.sid openTimeInstant, ids3.sid closeTimeInstant,
TO_TIMESTAMP_TZ(values1.value_name,''YYYY-MM-DD"T"HH24:MI:SSTZH:TZM'') openTime,
TO_TIMESTAMP_TZ(values2.value_name,''YYYY-MM-DD"T"HH24:MI:SSTZH:TZM'') closeTime
from ' || resource_id_map_view || ' values1,
' || resource_id_map_view || ' values2,
' || src_tab_view || ' ids1,
' || src_tab_view || ' ids2,
' || src_tab_view || ' ids3,
' || src_tab_view || ' ids4,
' || src_tab_view || ' ids5,
```

```
' || src_tab_view || ' ids5,
' || src_tab_view || ' ids6,
' || src_tab_view || ' ids7

where ids1.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND ids1.oid = ' || to_char(timePeriodClassId,'TM9') || '

AND ids2.pid = '|| to_char(rdfTypePropertyId,'TM9')||'
AND ids2.oid = '|| to_char(timeInstantClassId,'TM9') ||'
AND ids3.pid ='|| to_char(rdfTypePropertyId,'TM9')||'
AND ids3.oid = '|| to_char(timeInstantClassId,'TM9') ||'

AND ids4.sid = ids1.sid
AND ids4.pid = ' || to_char(openInstantPropertyId,'TM9') || '
AND ids4.oid = ids2.sid

AND ids5.sid = ids1.sid
AND ids5.pid = ' || to_char(closeInstantPropertyId,'TM9') || '
AND ids5.oid = ids3.sid

AND ids6.sid = ids2.sid
AND ids6.pid = ' || to_char (dateTimeValPropertyId,'TM9') || '
AND ids6.oid = values1.value_id

AND ids7.sid = ids3.sid
AND ids7.pid = ' || to_char (dateTimeValPropertyId,'TM9') || '
AND ids7.oid = values2.value_id

AND not exists
(select 1
from ' || src_tab_view || '
where sid = ids1.sid
AND pid = ' || to_char(timeDurationPropertyId,'TM9') || ')

AND not exists
(select 1
from ' || output_tab || '
where sid = ids1.sid
AND pid = ' || to_char(timeDurationPropertyId,'TM9') || ')';

-- compute the difference (in days) between the two Oracle
-- timestamps from our sqlStmt query. Store the days as
-- xsd:decimal.
insertStmt :=
'insert  into ' || output_tab || ' (sid, pid, o)
select timePeriodInstance,
' || to_char(timeDurationPropertyId,'TM9') || ',
''"'' || days || ''"^^xsd:decimal''
from (
select timePeriodInstance,
```

```
( trunc (
        (extract(day from (closeTime - openTime)) +
         extract (hour from (closeTime - openTime))/24 ), 1
        )
) days
from (' || sqlStmt || '))';
-- execute the query
execute immediate insertStmt;
-- commit our changes
commit;
end if;

optimization_flag := SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_NEWDATA_ONLY +
SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_UNIQDATA_ONLY;
-- return true to indicate success
return true;
-- handle any exceptions
exception
when others then
diag_message := 'error occurred: ' || SQLERRM;
return false;
end sem_inf_timeDuration;
```

Appendix 12-Definitions of Allen's Temporal Relations Based on Endpoint Relations

```
--the definition of basic relations in terms of endpoint relations

--before
INSERT INTO mdsys.semr_allenBasic VALUES(
 'defBefore',
 '(?t1 rdf:type :Time_Period) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
  (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) ',
 '(cv1 < ov2)',
 '(?t1 :before ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--meet
INSERT INTO mdsys.semr_allenBasic VALUES(
 'defMeet',
 '(?t1 rdf:type :Time_Period) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
  (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) ',
  '(cv1 = ov2)',
 '(?t1 :meet ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--during
INSERT INTO mdsys.semr_allenBasic VALUES(
 'defDuring',
 '(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?t1 :close_instant ?closeInstant1) (?openInstant1 :dateTimeValue ?ov1) (?closeInstant1 :dateTimeValue ?cv1)
  (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?t2 :close_instant ?closeInstant2) (?openInstant2 :dateTimeValue ?ov2) (?closeInstant2 :dateTimeValue ?cv2)',
  '((ov1 > ov2) and (cv1 < cv2))',
 '(?t1 :during ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--overlap
INSERT INTO mdsys.semr_allenBasic VALUES(
 'defOverLap',
 '(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?t1 :close_instant ?closeInstant1) (?openInstant1 :dateTimeValue ?ov1) (?closeInstant1 :dateTimeValue ?cv1)
  (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?t2 :close_instant ?closeInstant2) (?openInstant2 :dateTimeValue ?ov2) (?closeInstant2 :dateTimeValue ?cv2)',
  '((ov1 < ov2) and (cv1 > ov2) and (cv1 < cv2))',
 '(?t1 :overlap ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--equal
INSERT INTO mdsys.semr_allenBasic VALUES(
 'defEqual',
 '(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?t1 :close_instant ?closeInstant1) (?openInstant1 :dateTimeValue ?ov1) (?closeInstant1 :dateTimeValue ?cv1)
  (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?t2 :close_instant ?closeInstant2) (?openInstant2 :dateTimeValue ?ov2) (?closeInstant2 :dateTimeValue ?cv2)',
  '((ov1 = ov2) and (cv1 = cv2) and (t1 != t2))',
 '(?t1 :equal ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

```

```
--start
INSERT INTO mdsys.semr_allenBasic VALUES(
'defStart',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?t1 :close_instant ?closeInstant1) (?openInstant1 :dateTimeValue ?ov1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?t2 :close_instant ?closeInstant2) (?openInstant2 :dateTimeValue ?ov2) (?closeInstant2 :dateTimeValue ?cv2)',
 '((ov1 = ov2) and (cv1 < cv2))',
'(?t1 :start ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--finish
INSERT INTO mdsys.semr_allenBasic VALUES(
'defFinish',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?t1 :close_instant ?closeInstant1) (?openInstant1 :dateTimeValue ?ov1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?t2 :close_instant ?closeInstant2) (?openInstant2 :dateTimeValue ?ov2) (?closeInstant2 :dateTimeValue ?cv2)',
 '((ov1 > ov2) and (cv1 = cv2))',
'(?t1 :finish ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--the definition of indefinite fuzzy relations

--b_m_o
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_b_m_o',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?openInstant1 :dateTimeValue ?ov1) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) (?t2 :close_instant ?closeInstant2) (?closeInstant2 :dateTimeValue ?cv2)',
'((ov1 < ov2) and (cv1 < cv2))',
'(?t1 :before_or_meet_or_overlap ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--b_m_o_fi_di
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_b_m_o_fi_di',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?openInstant1 :dateTimeValue ?ov1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) ',
'(ov1 < ov2)',
'(?t1 :before_or_meet_or_overlap_or_finished_by_or_contain ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--b_m_o_s_d
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_b_m_o_s_d',
'(?t1 rdf:type :Time_Period) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :close_instant ?closeInstant2) (?closeInstant2 :dateTimeValue ?cv2) ',
'(cv1 < cv2)',
'(?t1 :before_or_meet_or_overlap_or_start_or_during ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

```

```
--concur
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'defConcur',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?openInstant1 :dateTimeValue ?ov1) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) (?t2 :close_instant ?closeInstant2) (?closeInstant2 :dateTimeValue ?cv2)',
'((ov1 < cv2) and (cv1 > ov2))',
'(?t1 :overlap_or_finished_by_or_contain_or_start_or_equal_or_started_by_or_during_or_finish_or_overlapped_by ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--d_f_oi
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_d_f_oi',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?openInstant1 :dateTimeValue ?ov1) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) (?t2 :close_instant ?closeInstant2) (?closeInstant2 :dateTimeValue ?cv2) ',
'((ov1 > ov2) and (ov1 < cv2))',
'(?t1 :during_or_finish_or_overlapped_by ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--d_o_s
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_d_o_s',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?openInstant1 :dateTimeValue ?ov1) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) (?t2 :close_instant ?closeInstant2) (?closeInstant2 :dateTimeValue ?cv2) ',
'((cv1 < cv2) and (cv1 > ov2))',
'(?t1 :during_or_overlap_or_start ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--f_eq_fi
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_f_eq_fi',
'(?t1 rdf:type :Time_Period) (?t1 :close_instant ?closeInstant1) (?closeInstant1 :dateTimeValue ?cv1)
 (?t2 rdf:type :Time_Period) (?t2 :close_instant ?closeInstant2) (?closeInstant2 :dateTimeValue ?cv2) ',
'((cv1 = cv2) and (t1 != t2))',
'(?t1 :finish_or_equal_or_finished_by ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--si_eq_s
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'def_si_eq_s',
'(?t1 rdf:type :Time_Period) (?t1 :open_instant ?openInstant1) (?openInstant1 :dateTimeValue ?ov1)
 (?t2 rdf:type :Time_Period) (?t2 :open_instant ?openInstant2) (?openInstant2 :dateTimeValue ?ov2) ',
'((ov1 = ov2) and (t1 != t2))',
'(?t1 :started_by_or_equal_or_start ?t2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

```

Appendix 13-Temporal Relation Reasoning Rules (Part)

```
--rules for allen's basic relations

--rules for before
--meet o meet
INSERT INTO mdsys.semr_allenBasic VALUES(
'before1',
'(?t1 :meet ?t2) (?t2 :meet ?t3)',
null,
'(?t1 :before ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--overlap o before
INSERT INTO mdsys.semr_allenBasic VALUES(
'before2',
'(?t1 :overlap ?t2) (?t2 :before ?t3)',
null,
'(?t1 :before ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--equal o before
INSERT INTO mdsys.semr_allenBasic VALUES(
'before3',
'(?t1 :equal ?t2) (?t2 :before ?t3)',
null,
'(?t1 :before ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--during o before
INSERT INTO mdsys.semr_allenBasic VALUES(
'before4',
'(?t1 :during ?t2) (?t2 :before ?t3)',
null,
'(?t1 :before ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--before o overlap
INSERT INTO mdsys.semr_allenBasic VALUES(
'before5',
'(?t1 :before ?t2) (?t2 :overlap ?t3)',
null,
'(?t1 :before ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

```
--rules for allen's fuzzy relations

--rules for b_m_o
--overlap o finished_by
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'b_m_o1',
'(?t1 :overlap ?t2) (?t2 :finished_by ?t3)',
null,
'(?t1 :before_or_meet_or_overlap ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--start o finished_by
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'b_m_o2',
'(?t1 :start ?t2) (?t2 :finished_by ?t3)',
null,
'(?t1 :before_or_meet_or_overlap ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--start o overlap
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'b_m_o3',
'(?t1 :start ?t2) (?t2 :overlap ?t3)',
null,
'(?t1 :before_or_meet_or_overlap ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

--overlap o overlap
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'b_m_o4',
'(?t1 :overlap ?t2) (?t2 :overlap ?t3)',
null,
'(?t1 :before_or_meet_or_overlap ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));


--rules for b_m_o_fi_di
--contain o before
INSERT INTO mdsys.semr_allenFuzzy VALUES(
'b_m_o_fi_dil',
'(?t1 :contain ?t2) (?t2 :before ?t3)',
null,
'(?t1 :before_or_meet_or_overlap_or_finished_by_or_contain ?t3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

Appendix 14- Subset Relationship Reasoning Rules between Temporal Relations (Part)

```
INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp1',
'(?r1 owl:equivalentProperty :before) (?r2 owl:equivalentProperty :before_or_meet_or_overlap)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp2',
'(?r1 owl:equivalentProperty :after) (?r2 owl:equivalentProperty :after_or_met_by_or_overlapped_by)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp29',
'(?r1 owl:equivalentProperty :before_or_meet_or_overlap) (?r2 owl:equivalentProperty :before_or_meet_or_overlap_or_finished_by_or_contain)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp30',
'(?r1 owl:equivalentProperty :before_or_meet_or_overlap) (?r2 owl:equivalentProperty :before_or_meet_or_overlap_or_start_or_during)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp39',
'(?r1 owl:equivalentProperty :during_or_overlap_or_start) (?r2 owl:equivalentProperty :before_or_meet_or_overlap_or_start_or_during)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp40',
'(?r1 owl:equivalentProperty :during_or_overlap_or_start)
 (?r2 owl:equivalentProperty :overlap_or_finished_by_or_contain_or_start_or_equal_or_started_by_or_during_or_finish_or_overlapped_by)',
null,
'(?r1 :subSetOf ?r2)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationSubSetProp VALUES(
'subSetProp45',
'(?r1 :subSetOf ?r2) (?r2 :subSetOf ?r3)',
null,
'(?r1 :subSetOf ?r3)',
SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

Appendix 15-Intersection Reasoning Rules between Temporal Relations (Part)

```
INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
 'relIntersection1',
 '(?t1 :contain_or_finished_by_or_overlap ?t2) (?t1 :contain_or_overlapped_by_or_started_by ?t2)',
 null,
 '(?t1 :contain ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
 'relIntersection2',
 '(?t1 :contain_or_finished_by_or_overlap ?t2) (?t1 :during_or_overlap_or_start ?t2)',
 null,
 '(?t1 :overlap ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
 'relIntersection3',
 '(?t1 :contain_or_finished_by_or_overlap ?t2) (?t1 :finish_or_equal_or_finished_by ?t2)',
 null,
 '(?t1 :finished_by ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
 'relIntersection4',
 '(?t1 :contain_or_finished_by_or_overlap ?t2) (?t1 :after_or_met_by_or_overlapped_by_or_started_by_or_contain ?t2)',
 null,
 '(?t1 :contain ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
 'relIntersection25',
 '(?t1 :after_or_met_by_or_overlapped_by_or_finish_or_during ?t2) (?t1 :after_or_met_by_or_overlapped_by_or_started_by_or_contain ?t2)',
 null,
 '(?t1 :after_or_met_by_or_overlapped_by ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
 'relIntersection31',
 '(?t1 :before_or_meet_or_overlap_or_finished_by_or_contain ?t2) (?t1 :before_or_meet_or_overlap_or_start_or_during ?t2)',
 null,
 '(?t1 :before_or_meet_or_overlap ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));

INSERT INTO mdsys.semr_allenRelationIntersection VALUES(
 'relIntersection33',
 '(?t1 :before_or_meet_or_overlap_or_start_or_during ?t2)
  (?t1 :overlap_or_finished_by_or_contain_or_start_or_equal_or_started_by_or_during_or_finish_or_overlapped_by ?t2)',
 null,
 '(?t1 :during_or_overlap_or_start ?t2)',
 SEM_ALIASES(SEM_ALIAS('','http://www.usyd.edu.au/hitru/antibiotics#')));
```

Appendix 16-Temporal Relation Inconsistency Checking Inference Function (Part)

```
create or replace
function sem_inf_consistencyChecking (
src_tab_view in varchar2,
resource_id_map_view in varchar2,
output_tab in varchar2,
action in varchar2,
num_calls in number,
tplInferredLastRound in number,
options in varchar2 default null,
optimization_flag out number,
diag_message out varchar2 )
return boolean
as
timePeriodClassId number;
rdfTypePropertyId number;

beforePropertyId number;
afterPropertyId number;
meetPropertyId number;
metByPropertyId number;
overlapPropertyId number;
overlappedByPropertyId number;
startPropertyId number;
startedByPropertyId number;
equalPropertyId number;
duringPropertyId number;
containPropertyId number;
finishPropertyId number;
finishedByPropertyId number;

b_m_oPropertyId number;
bi_mi_oiPropertyId number;
b_m_o_fi_diPropertyId number;
bi_mi_oi_f_dPropertyId number;
b_m_o_s_dPropertyId number;
bi_mi_oi_si_diPropertyId number;
concurPropertyId number;
d_f_oiPropertyId number;
di_fi_oPropertyId number;
d_o_sPropertyId number;
di_oi_siPropertyId number;
f_eq_fiPropertyId number;
si_eq_sPropertyId number;

hasConfRelPropertyId number;
confilctPropertyId number;

```

```
48  --conflict between basic and basic or between basic and fuzzy
49  sqlStmt1 varchar2(4000);
50  insertStmt1 varchar2(4000);
51  sqlStmt2 varchar2(4000);
52  insertStmt2 varchar2(4000);
53  sqlStmt3 varchar2(4000);
54  insertStmt3 varchar2(4000);
55  sqlStmt4 varchar2(4000);
56  insertStmt4 varchar2(4000);
57  sqlStmt5 varchar2(4000);
58  insertStmt5 varchar2(4000);
59  sqlStmt6 varchar2(4000);
60  insertStmt6 varchar2(4000);
61  sqlStmt7 varchar2(4000);
62  insertStmt7 varchar2(4000);
63  sqlStmt8 varchar2(4000);
64  insertStmt8 varchar2(4000);
65  sqlStmt9 varchar2(4000);
66  insertStmt9 varchar2(4000);
67  sqlStmt10 varchar2(4000);
68  insertStmt10 varchar2(4000);
69  sqlStmt11 varchar2(4000);
70  insertStmt11 varchar2(4000);
71  sqlStmt12 varchar2(4000);
72  insertStmt12 varchar2(4000);
73  sqlStmt13 varchar2(4000);
74  insertStmt13 varchar2(4000);
75
76  --conflict between fuzzy relation and fuzzy relation
77  sqlStmt14 varchar2(4000);
78  insertStmt14 varchar2(4000);
79  sqlStmt15 varchar2(4000);
80  insertStmt15 varchar2(4000);
81  sqlStmt16 varchar2(4000);
82  insertStmt16 varchar2(4000);
83  sqlStmt17 varchar2(4000);
84  insertStmt17 varchar2(4000);
85  sqlStmt18 varchar2(4000);
86  insertStmt18 varchar2(4000);
87  sqlStmt19 varchar2(4000);
88  insertStmt19 varchar2(4000);
89  sqlStmt20 varchar2(4000);
90  insertStmt20 varchar2(4000);
91  sqlStmt21 varchar2(4000);
92  insertStmt21 varchar2(4000);
93  sqlStmt22 varchar2(4000);
94  insertStmt22 varchar2(4000);
```

```
insertStmt23 varchar2(4000);

pragma autonomous_transaction;
begin
if (action = 'RUN') then
timePeriodClassId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#Time_Period');
rdfTypePropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.w3.org/1999/02/22-rdf-syntax-ns#type');

beforePropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#before');
afterPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#after');
meetPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#meet');
metByPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#met_by');
overlapPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#overlap');
overlappedByPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#overlapped_by');
startPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#start');
startedByPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#started_by');
equalPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#equal');
duringPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#during');
containPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#contain');
finishPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#finish');
finishedByPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#finished_by');

b_m_oPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#before_or_meet_or_overlap');
bi_mi_oiPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#after_or_met_by_or_overlapped_by');
b_m_o_fi_diPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#before_or_meet_or_overlap_or_finished_by_or_contain');
bi_mi_oi_f_dPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#after_or_met_by_or_overlapped_by_or_finish_or_during');
b_m_o_s_dPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#before_or_meet_or_overlap_or_start_or_during');
bi_mi_oi_si_diPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#after_or_met_by_or_overlapped_by_or_started_by_or_contain');
concurPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#overlap_or_finished_by_or_contain_or_start_or_equal_or_started_by_or_during_or_finish_or_overlapped_by');
d_f_oiPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#during_or_finish_or_overlapped_by');
di_fi_oPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#contain_or_finished_by_or_overlap');
d_o_sPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#during_or_overlap_or_start');
di_oi_siPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#contain_or_overlapped_by_or_started_by');
f_eq_fiPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#finish_or_equal_or_finished_by');
si_eq_sPropertyId := sdo_sem_inference.oracle_orardf_res2vid('http://www.usyd.edu.au/hitru/antibiotics#started_by_or_equal_or_start');

hasConfRelPropertyId := sdo_sem_inference.oracle_orardf_add_res('http://www.usyd.edu.au/hitru/antibiotics#has_conflict_temporal_relation');
confilctPropertyId := sdo_sem_inference.oracle_orardf_add_res('http://www.usyd.edu.au/hitru/antibiotics#conflict');

```

```
--conflict between basic relation and basic relation or between basic relation and fuzzy relation
--inconsistencyRule 1: before and its disjoint relations

sqlStmt1 :=
'select ids1.sid dose1, ids2.sid dose2, ids3.pid timeRel, ids4.pid conflictTimeRel
from
' || src_tab_view || ' ids1,
' || src_tab_view || ' ids2,
' || src_tab_view || ' ids3,
' || src_tab_view || ' ids4

where ids1.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND ids1.oid = ' || to_char(timePeriodClassId,'TM9') || '

AND ids2.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
AND ids2.oid = ' || to_char(timePeriodClassId,'TM9') || '

AND ids3.sid = ids1.sid
AND ids3.pid = ' || to_char(beforePropertyId ,'TM9') || '
AND ids3.oid = ids2.sid

AND ids4.sid = ids1.sid
AND ((ids4.pid = '|| to_char(afterPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(meetPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(metByPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(overlapPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(overlappedByPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(startPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(startedByPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(equalPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(duringPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(containPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(finishPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(finishedByPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(bi_mi_oiPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(bi_mi_oi_f_dPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(bi_mi_oi_si_diPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(concurPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(d_f_oiPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(di_fi_oPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(d_o_sPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(di_oi_siPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(f_eq_fiPropertyId,'TM9') || ') OR
     (ids4.pid = '|| to_char(si_eq_sPropertyId,'TM9') || ')
     )
AND ids4.oid = ids2.sid
';
```

```
182
183  insertStmt1 :=
184  'insert  into ' || output_tab || ' (sid, pid, oid)
185  select timeRel,
186  ' || to_char(confilctPropertyId,'TM9') || ',
187  conflictTimeRel
188  from (' || sqlStmt1 || ')
189  UNION
190  select dose1,
191  ' || to_char(hasConfRelPropertyId,'TM9') || ',
192  dose2
193  from (' || sqlStmt1 || ')
194  ';
195
196  --inconsistencyRule 2: after and its disjoint relations
197
198  sqlStmt2 :=
199  'select ids1.sid dose1, ids2.sid dose2, ids3.pid timeRel, ids4.pid conflictTimeRel
200  from
201  ' || src_tab_view || ' ids1,
202  ' || src_tab_view || ' ids2,
203  ' || src_tab_view || ' ids3,
204  ' || src_tab_view || ' ids4
205
206  where ids1.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
207  AND ids1.oid = ' || to_char(timePeriodClassId,'TM9') || '
208
209  AND ids2.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
210  AND ids2.oid = ' || to_char(timePeriodClassId,'TM9') || '
211
212  AND ids3.sid = ids1.sid
213  AND ids3.pid = ' || to_char(afterPropertyId ,'TM9') || '
214  AND ids3.oid = ids2.sid
215
216  AND ids4.sid = ids1.sid
217  AND ((ids4.pid = '|| to_char(meetPropertyId,'TM9') ||') OR
218       (ids4.pid = '|| to_char(metByPropertyId,'TM9') ||') OR
219       (ids4.pid = '|| to_char(overlapPropertyId,'TM9') ||') OR
220       (ids4.pid = '|| to_char(overlappedByPropertyId,'TM9') || ') OR
221       (ids4.pid = '|| to_char(startPropertyId,'TM9') || ') OR
222       (ids4.pid = '|| to_char(startedByPropertyId,'TM9') || ') OR
223       (ids4.pid = '|| to_char(equalPropertyId,'TM9') || ') OR
224       (ids4.pid = '|| to_char(duringPropertyId,'TM9') || ') OR
225       (ids4.pid = '|| to_char(containPropertyId,'TM9') || ') OR
226       (ids4.pid = '|| to_char(finishPropertyId,'TM9') || ') OR
227       (ids4.pid = '|| to_char(finishedByPropertyId,'TM9') || ') OR
228       (ids4.pid = '|| to_char(b_m_oPropertyId,'TM9') || ') OR
229       (ids4.pid = '|| to_char(b_m_o_fi_diPropertyId,'TM9') || ') OR
230       (ids4.pid = '|| to_char(b_m_o_s_dPropertyId,'TM9') || ') OR
```

```
1194  --inconsistencyRule 23: di_oi_si and its disjoint relations
1195
1196  sqlStmt23 :=
1197  'select idsl.sid dosel, ids2.sid dose2, ids3.pid timeRel, ids4.pid conflictTimeRel
1198  from
1199  ' || src_tab_view || ' idsl,
1200  ' || src_tab_view || ' ids2,
1201  ' || src_tab_view || ' ids3,
1202  ' || src_tab_view || ' ids4
1203
1204  where idsl.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
1205  AND idsl.oid = ' || to_char(timePeriodClassId,'TM9') || '
1206
1207  AND ids2.pid = ' || to_char(rdfTypePropertyId,'TM9') || '
1208  AND ids2.oid = ' || to_char(timePeriodClassId,'TM9') || '
1209
1210  AND ids3.sid = idsl.sid
1211  AND ids3.pid = ' || to_char(di_oi_siPropertyId, 'TM9') || '
1212  AND ids3.oid = ids2.sid
1213
1214  AND ids4.sid = idsl.sid
1215  AND ids4.pid = '|| to_char(f_eq_fiPropertyId,'TM9') || '
1216  AND ids4.oid = ids2.sid
1217  ';
1218
1219  insertStmt23 :=
1220  'insert  into ' || output_tab || ' (sid, pid, oid)
1221  select timeRel,
1222  ' || to_char(confilctPropertyId,'TM9') || ',
1223  conflictTimeRel
1224  from (' || sqlStmt23 || ')
1225  UNION
1226  select dosel,
1227  ' || to_char(hasConfRelPropertyId,'TM9') || ',
1228  dose2
1229  from (' || sqlStmt23 || ')
1230  ';
1231
1232  execute immediate insertStmt1;
1233  execute immediate insertStmt2;
1234  execute immediate insertStmt3;
1235  execute immediate insertStmt4;
1236  execute immediate insertStmt5;
1237  execute immediate insertStmt6;
1238  execute immediate insertStmt7;
1239  execute immediate insertStmt8;
1240  execute immediate insertStmt9;
1241  execute immediate insertStmt10;
```

```
1242  execute immediate insertStmt11;
1243  execute immediate insertStmt12;
1244  execute immediate insertStmt13;
1245  execute immediate insertStmt14;
1246  execute immediate insertStmt15;
1247  execute immediate insertStmt16;
1248  execute immediate insertStmt17;
1249  execute immediate insertStmt18;
1250  execute immediate insertStmt19;
1251  execute immediate insertStmt20;
1252  execute immediate insertStmt21;
1253  execute immediate insertStmt22;
1254  execute immediate insertStmt23;
1255
1256  commit;
1257  end if;
1258  optimization_flag := SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_NEWDATA_ONLY +
1259  SDO_SEM_INFERENCE.INF_EXT_OPT_FLAG_UNIQDATA_ONLY;
1260  return true;
1261  exception
1262  when others then
1263  diag_message := 'error occurred: ' || SQLERRM;
1264  return false;
1265  end sem_inf_consistencyChecking;
```

Appendix 17- System Outputs Based on Real Patient Data (Part)

Medical Case 1 in Community Acquired Pneumonia Category: Patient who has community acquired pneumonia, but has not had severe sepsis and penicillin hypersensitivity

Recommended regimens (Q1):						
Patient Name	**Recommended Regimen**	**Dose Agent**	**Dose Amount**	**Dose Interval**	**Administration Route**	**Note**
Patient 202	Medication 1, 2 and 3 or Medication 2 and 5 (Community Acquired Pneumonia)	Benzylpenicillin, Azithromycin and Gentamicin OR Azithromycin and Ceftriaxone	null	null	null	Please refer to medication 1, 2, 3 and 5 (community acquired pneumonia) for the details of dose amount, dose interval and administration route.
Patient 253	Medication 1, 2 and 3 or Medication 2 and 5 (Community Acquired Pneumonia)	Benzylpenicillin, Azithromycin and Gentamicin OR Azithromycin and Ceftriaxone	null	null	null	Please refer to medication 1, 2, 3 and 5 (community acquired pneumonia) for the details of dose amount, dose interval and administration route.
Patient 368	Medication 1, 2 and 3 or Medication 2 and 5 (Community Acquired Pneumonia)	Benzylpenicillin, Azithromycin and Gentamicin OR Azithromycin and Ceftriaxone	null	null	null	Please refer to medication 1, 2, 3 and 5 (community acquired pneumonia) for the details of dose amount, dose interval and administration route.

Patient 425	Medication 1, 2 and 3 or Medication 2 and 5 (Community Acquired Pneumonia)	Benzylpenicillin, Azithromycin and Gentamicin OR Azithromycin and Ceftriaxone	null	null	null	Please refer to medication 1, 2, 3 and 5 (community acquired pneumonia) for the details of dose amount, dose interval and administration route.

Patient's administered antibiotics which are recommended by the guideline (Q2):

Patient Name	Administered Drug	Dose Amount	Route of Administration
Patient 253	azithromycin3218558	250 mg	PO
Patient 253	ceftriaxone3216329	1 g	IV
Patient 253	ceftriaxone3218557	1 g	IV
Patient 368	ceftriaxone659664	1 g	IV

Patient's administered antibiotics which are not recommended by the guideline (Q2):

Patient Name	Administered Drug	Dose Amount	Route of Administration
Patient 202	cefuroxime2020001	250 mg	IV
Patient 202	clindamycin2020001	600 mg	IV
Patient 253	cefazolin3214543	1 g	IV
Patient 253	erythromycin2530001	null	null
Patient 253	levofloxacin3214920	500 mg	IV
Patient 253	vancomycin3225569	1000 mg	IV
Patient 368	levofloxacin659885	250 mg	PO
Patient 368	metronidazole659663	500 mg	IV
Patient 425	cefotaxime4250001	2 g	IV
Patient 425	erythromycin4250001	1 g	IV

Dose intervals of each administered antibiotic of patients (Q3):

Patient Name	Administered Drug	Interval Start	Interval End	Length (Hours)

Patient 202	cefuroxime2020001	2431-01-04T12:00:00	2431-01-04T20:00:00	8
Patient 202	cefuroxime2020001	2431-01-04T20:00:00	2431-01-05T04:00:00	8
Patient 202	clindamycin2020001	2431-01-04T12:00:00	2431-01-04T20:00:00	8
Patient 202	clindamycin2020001	2431-01-04T20:00:00	2431-01-05T04:00:00	8
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-13T15:00:00	24
Patient 253	azithromycin3218558	2944-01-13T15:00:00	2944-01-14T15:00:00	24
Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	20.5
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	23
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-13T15:00:00	24
Patient 253	ceftriaxone3218557	2944-01-13T15:00:00	2944-01-14T15:00:00	24
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	15
Patient 253	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T08:00:00	12
Patient 253	vancomycin3225569	2944-01-15T08:00:00	2944-01-15T20:00:00	12
Patient 368	ceftriaxone659664	2568-06-23T20:00:00	2568-06-24T09:00:00	13
Patient 368	levofloxacin659885	2568-06-23T22:00:00	2568-06-24T22:00:00	24
Patient 368	levofloxacin659885	2568-06-24T22:00:00	2568-06-25T22:00:00	24
Patient 368	levofloxacin659885	2568-06-25T22:00:00	2568-06-26T22:00:00	24
Patient 368	levofloxacin659885	2568-06-26T22:00:00	2568-06-27T22:00:00	24
Patient 368	levofloxacin659885	2568-06-27T22:00:00	2568-06-28T22:00:00	24
Patient 368	metronidazole659663	2568-06-23T20:00:00	2568-06-24T04:00:00	8
Patient 368	metronidazole659663	2568-06-24T04:00:00	2568-06-24T12:00:00	8
Patient 368	metronidazole659663	2568-06-24T12:00:00	2568-06-24T20:00:00	8

Patient 368	metronidazole659663	2568-06-24T20:00:00	2568-06-25T03:30:00	7.5
Patient 368	metronidazole659663	2568-06-25T03:30:00	2568-06-25T11:00:00	7.5
Patient 425	cefotaxime4250001	2431-06-21T14:00:00	2431-06-21T22:00:00	8
Patient 425	cefotaxime4250001	2431-06-21T22:00:00	2431-06-22T07:00:00	9
Patient 425	erythromycin4250001	2431-06-21T12:00:00	2431-06-21T18:00:00	6
Patient 425	erythromycin4250001	2431-06-21T18:00:00	2431-06-22T00:00:00	6
Patient 425	erythromycin4250001	2431-06-22T00:00:00	2431-06-22T06:00:00	6
Patient 425	erythromycin4250001	2431-06-22T06:00:00	2431-06-22T12:00:00	6

Dose intervals of patient's administered ceftriaxone which are not equal to the recommended 24 hours interval (Q3):

Patient Name	Administered Drug	Dose Time (Interval Start)	Dose Time (Interval End)	Interval Length (Hours)
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	23
Patient 368	ceftriaxone659664	2568-06-23T20:00:00	2568-06-24T09:00:00	13

Dose durations of each administered antibiotic of patients (Q3):

Patient Name	Administered Drug	Start Time	Finish Time	Duration Length (Days)
Patient 202	cefuroxime2020001	2431-01-04T12:00:00	2431-01-05T04:00:00	0.6
Patient 202	clindamycin2020001	2431-01-04T12:00:00	2431-01-05T04:00:00	0.6
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00	2
Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	0.8
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	0.9
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00	2
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	0.6
Patient	vancomycin3225569	2944-01-	2944-01-	1

253		14T20:00:00	15T20:00:00	
Patient 368	ceftriaxone659664	2568-06-23T20:00:00	2568-06-24T09:00:00	0.5
Patient 368	levofloxacin659885	2568-06-23T22:00:00	2568-06-28T22:00:00	5
Patient 368	metronidazole659663	2568-06-23T20:00:00	2568-06-25T11:00:00	1.6
Patient 425	cefotaxime4250001	2431-06-21T14:00:00	2431-06-22T07:00:00	0.7
Patient 425	erythromycin4250001	2431-06-21T12:00:00	2431-06-22T12:00:00	1

Dose duration compliance checking is not available in this medical case (Q3)

Basic temporal relations between patient's administered antibiotics (Q4):

Patient Name	Administered Drug	Start Time	Finish Time	Temporal Relation	Administered Drug	Start Time	Finish Time
Patient 202	cefuroxime2020001	2431-01-04T12:00:00	2431-01-05T04:00:00	equal	clindamycin2020001	2431-01-04T12:00:00	2431-01-05T04:00:00
Patient 202	clindamycin2020001	2431-01-04T12:00:00	2431-01-05T04:00:00	equal	cefuroxime2020001	2431-01-04T12:00:00	2431-01-05T04:00:00
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00	after	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00	after	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00	equal	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00	after	erythromycin2530001	2944-01-10T00:00:00	
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00	after	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00
Patient 253	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00	before	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00
Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	before	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	before	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00

Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	before	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	overlap	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00
Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	before	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	before	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	after	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	before	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	after	erythromycin2530001	2944-01-10T00:00:00	
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	after	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00
Patient 253	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00	before	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00	equal	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00	after	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00	after	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00	after	erythromycin2530001	2944-01-10T00:00:00	
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00	after	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00
Patient 253	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00	before	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00
Patient 253	erythromycin2530001	2944-01-10T00:00:00		before	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	erythromycin2530001	2944-01-10T00:00:00		before	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00

Patient 253	erythromycin 2530001	2944-01-10T00:00:00		before	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	before	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	overlapped by	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	before	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	before	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	before	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00
Patient 253	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00	after	azithromycin3218558	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00	after	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00
Patient 253	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00	after	ceftriaxone3216329	2944-01-11T15:00:00	2944-01-12T14:00:00
Patient 253	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00	after	ceftriaxone3218557	2944-01-12T15:00:00	2944-01-14T15:00:00
Patient 253	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00	after	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00
Patient 368	ceftriaxone659664	2568-06-23T20:00:00	2568-06-24T09:00:00	overlap	levofloxacin659885	2568-06-23T22:00:00	2568-06-28T22:00:00
Patient 368	ceftriaxone659664	2568-06-23T20:00:00	2568-06-24T09:00:00	start	metronidazole659663	2568-06-23T20:00:00	2568-06-25T11:00:00
Patient 368	levofloxacin659885	2568-06-23T22:00:00	2568-06-28T22:00:00	overlapped by	ceftriaxone659664	2568-06-23T20:00:00	2568-06-24T09:00:00
Patient 368	levofloxacin659885	2568-06-23T22:00:00	2568-06-28T22:00:00	overlapped by	metronidazole659663	2568-06-23T20:00:00	2568-06-25T11:00:00
Patient 368	metronidazole659663	2568-06-23T20:00:00	2568-06-25T11:00:00	started by	ceftriaxone659664	2568-06-23T20:00:00	2568-06-24T09:00:00
Patient 368	metronidazole659663	2568-06-23T20:00:00	2568-06-25T11:00:00	overlap	levofloxacin659885	2568-06-23T22:00:00	2568-06-28T22:00:00

Patient 425	cefotaxime4250001	2431-06-21T14:00:00	2431-06-22T07:00:00	during	erythromycin4250001	2431-06-21T12:00:00	2431-06-22T12:00:00
Patient 425	erythromycin4250001	2431-06-21T12:00:00	2431-06-22T12:00:00	contain	cefotaxime4250001	2431-06-21T14:00:00	2431-06-22T07:00:00

Indefinite fuzzy temporal relations between patient's administered antibiotics (Q4):

Patient Name	Administered Drug	Start Time	Finish Time	Temporal Relation	Administered Drug	Start Time	Finish Time
Patient 253	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00	bi_mi_oi_f_d	erythromycin2530001	2944-01-10T00:00:00	
Patient 253	erythromycin2530001	2944-01-10T00:00:00		b_m_o_fi_di	cefazolin3214543	2944-01-10T12:00:00	2944-01-11T08:30:00
Patient 253	erythromycin2530001	2944-01-10T00:00:00		b_m_o_fi_di	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00
Patient 253	erythromycin2530001	2944-01-10T00:00:00		b_m_o_fi_di	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00
Patient 253	levofloxacin3214920	2944-01-10T23:00:00	2944-01-11T14:00:00	bi_mi_oi_f_d	erythromycin2530001	2944-01-10T00:00:00	
Patient 253	vancomycin3225569	2944-01-14T20:00:00	2944-01-15T20:00:00	bi_mi_oi_f_d	erythromycin2530001	2944-01-10T00:00:00	

Inconsistent temporal relation found (Q5):

Nil

Bibliography

[1] Field, MJ & Lohr, KN, 1993, 'Guidelines for clinical practice: from development to use', *BMJ: British Medical Journal*, vol. 306, no. 6884, pp. 1077, viewed 1 September 2012, USYD databases and electronic resources, JSTOR.

[2] Rosenbrand, K, Croonenborg, JV & Wittenberg, J, 2008, 'Guideline Development' in Teije, AT, Miksch, S & Lucas, P (eds), *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, IOS Press, Amsterdam, pp. 5.

[3] Shahar, Y, 2002, 'Automated Support to Clinical Guidelines and Care Plans: The Intention-Oriented View', viewed 7 September 2012, <http://www.openclinical.org/briefingpaperShahar.html>.

[4] De Clercq, P, Kaiser, K & Hasman A, 2008, 'Computer-interpretable Guideline Formalisms' in Teije, AT, Miksch, S & Lucas, P (eds), *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, IOS Press, Amsterdam, pp. 22-43.

[5] Wang, D, Peleg, M, Tu, SM, Shortliffe, EH & Greenes, RA, 2001, 'Representation of Clinical Practice Guidelines for Computer-Based Implementations', *Medinfo*, vol. 84, pp. 285 – 289.

[6] W3C, 2007, *Representing Clinical Guidelines and Protocols on a Semantic Web Framework*, viewed 10 October 2012, <http://www.w3.org/wiki/HCLS/ACPPTaskForce>.

[7] Honderich, T (ed), 2005, *The Oxford Companion to Philosophy*, 2nd edn, Oxford University Press, Landon, pp. 670.

[8] Studer, R, Benjamins, VR & Fensel, D, 1998, 'Knowledge Engineering: Principles and Methods,' *Data & Knowledge Engineering*, vol. 25, pp. 161-197, viewed 15 November 2012,

USYD databases and electronic resources, ScienceDirect, DOI: 10.1016/S0169-023X(97)00056-6.

[9] W3C, 2004, *RDF Prime*, viewed 20 September 2012, <http://www.w3.org/TR/rdf-primer/#basicconcepts>.

[10] W3C, 2013, *RDF1.1 Concepts and Abstract Syntax*, viewed 20 September 2012, <http://www.w3.org/TR/rdf11-concepts/#resources-and-statements>.

[11] W3C, 2004, *RDF Vocabulary Description Language 1.0: RDF Schema*, viewed 22 September 2012, <http://www.w3.org/TR/rdf-schema/#ch_member>.

[12] Antoniou, G & Harmelen, FV, 2008, *A Semantic Web Primer*, 2nd edn, The MIT Press, Cambridge, Massachusetts, pp.116.

[13] W3C, 2012, *OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax*, 2nd edn, viewed 28 September 2012, < http://www.w3.org/TR/owl2-syntax/>.

[14] Krötzsch, M, Simancík, F, & Ian Horrocks, I, 2012, *A Description Logic Primer*, CoRR, viewed 5 October 2012, < http://arxiv.org/pdf/1201.4089.pdf >.

[15] W3C, 2007, *Representing Clinical Guidelines and Protocols on a Semantic Web Framework*, viewed 10 October 2012, <http://www.w3.org/wiki/HCLS/ACPPTaskForce>.

[16] Kashyap, V, Alfredo Morales, A & Hongsermeier, T, 2005, *Creation and maintenance of implementable clinical guideline specifications*, viewed 15 October 2012, <www.w3.org/2005/04/swls/clinical_guidelines.pdf>.

[17] De Clercq, PA, 2000, 'An Ontological Approach for the Development of Shareable Guidelines', *Proceedings of the AMIA Symposium*, pp. 166-70, viewed 1 December 2012, PMC database.

[18] Casteleiro, MA & Des Diz, JJ, 2008, 'Clinical practice guidelines: A case study of combining OWL-S, OWL, and SWRL', *Knowledge-Based Systems*, vol. 21, issue 3, pp. 247-255, viewed 5 November 2012, USYD databases and electronic resources, ScienceDirect, DOI:10.1016/j.knosys.2007.11.008.

[19] Chen, R, Bau, C & Huan, Y, 2010, 'Development of Anti-Diabetic Drugs Ontology for Guideline-Based Clinical Drugs Recommend System Using OWL and SWRL', *Fuzzy Systems (FUZZ), 2010 IEEE International Conference,* viewed 10 November 2012, USYD databases and electronic resources, IEEE Xplore, DOI: 10.1109/FUZZY.2010.5584139.

[20] Abidi, SR, 2007, 'Ontology-based Modelling of Breast Cancer Follow-up Clinical Practice Guideline for Providing Clinical Decision Support', *Twentieth IEEE International Symposium on Computer-Based Medical Systems*, viewed 12 November 2012, USYD databases and electronic resources, IEEE Xplore, DOI: 10.1109/CBMS.2007.80.

[21] Martínez- Romero, M, Vázquez-Naya, JM, Pereira, J, Pazos, A, Pereira, M & Baño, G, 2012, 'An Ontology-based Expert System for Decision Support in Cardiac Intensive Care Environments', in Grana, M, Toro, C, Posada, J, Howlett, RJ & Jain, LC (eds), *Advances in Knowledge-Based and Intelligent Information and Engineering Systems*, IOS Press, Amsterdam, pp. 1360 – 1369.

[22] Abidi, S, Cox, J, Abidi, SSR & Shepherd, M, 2012, 'Using OWL Ontologies for Clinical Guidelines Based Comorbid Decision Support', *2012 45th Hawaii International Conference on System Sciences,* viewed 15 November 2012, USYD databases and electronic resources, IEEE Xplore, DOI: 10.1109/HICSS.2012.629.

[23] Welty, C & Fikes, R, 2006, 'A Reusable Ontology for Fluents in OWL', *Proceedings of FOIS*, pp. 226-236, viewed 20 February 2013, CiteSeerX database.

[24] O'Connor, MJ & Das, AK, 2011, 'A Method for Representing and Querying Temporal Information in OWL', *Third International Joint Conference, Biomedical Engineering Systems*

and Technologies, vol. 127, pp. 97-110, viewed 2 December 2012, USYD databases and electronic resources, SpringerLink.

[25] Artale A & Franconi, E, 2000, 'A survey of temporal extensions of description logics', *Annals of Mathematics and Artificial Intelligence,* vol. 30, pp.171-210, viewed 28 December 2012, USYD databases and electronic resources, SpringerLink.

[26] Milea V, Frasincar F & Kaymak U, 2012, 'tOWL: A Temporal Web Ontology Language', *IEEE Transactions on Systems, Man, and Cybernetics*, *Part B (*Cybernetics*),* vol. 42, pp.268-281, viewed 15 January 2013, USYD databases and electronic resources, IEEE Xplore, DOI:10.1109/TSMCB.2011.2162582.

[27] W3C, 2006, *Defining N-ary Relations on the Semantic Web*, viewed 30 September 2012, <http://www.w3.org/TR/2006/NOTE-swbp-n-aryRelations-20060412/>.

[28] Shankar, RD, Martins, SB, O'Connor, MJ & Das, AK, 2008, 'An ontological approach to representing and reasoning with temporal constraints in clinical trial protocols', *2008 International Conference on Health Informatics (HEALTHINF 2008)*, viewed 20 February 2013, USYD databases and electronic resources, ProQuest database.

[29] Tao C, Wei W, Solbrig HR, Savova G & Chute CG, 2010, 'CNTRO: A Semantic Web Ontology for Temporal Relation Inferencing in Clinical Narratives', *AMIA Annu Symp Proc. 2010*, pp. 787–791, viewed 2 April 2013, <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041418/>.

[30] Batsakis, S & Petrakis, EGM, 2011, 'Representing Temporal Knowledge in the Semantic Web: The Extended 4D Fluents Approach', *Combinations of Intelligent Methods and Applications Smart Innovation, Systems and Technologies*, vol. 8, 2011, pp. 55-69, viewed 15 March 2013, USYD databases and electronic resources, SpringerLink.

[31] Sider, T, 2001, 'Four Dimensionalism', Oxford University Press, vol. 40, pp. 380 – 394, viewed 13 March 2013, USYD databases and electronic resources, Wiley Online Library, DOI: 10.1111/j.0029-4624.2006.00617.x.

[32] W3C, 2006, *Time Ontology in OWL*, viewed 3 July 2012, <http://www.w3.org/TR/owl-time/>.

[33] Krieger, HU, Kiefer, B & Declerck, 2008, T 'A Framework for Temporal Representation and Reasoning in Business Intelligence Applications', *AI Meets Business Rules and Process Management-Papers from AAAI 2008 Spring Symposium*, vol. SS-08-01, pp. 59-70, viewed 5 April 2013, <http://www.aaai.org/Library/Symposia/Spring/2008/ss08-01-008.php>.

[34] Okeyo, G, Chen, L, Wang, H & Roy, S, 2012, 'A Hybrid Ontological and Temporal Approach for Composite Activity Modelling', *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference*, pp. 1763-1770, viewed 20 March 2013, USYD databases and electronic resources, IEEE Xplore, DOI: 10.1109/TrustCom.2012.34.

[35] Krieger, HU, Kiefer, B & Declerck, 2008, T 'A Framework for Temporal Representation and Reasoning in Business Intelligence Applications', *AI Meets Business Rules and Process Management-Papers from AAAI 2008 Spring Symposium,* vol. SS-08-01, pp. 59-70, viewed 5 April 2013, <http://www.aaai.org/Library/Symposia/Spring/2008/ss08-01-008.php>.

[36] Harbelot, B, Arenas, H & Cruz, C, 2013, 'A Semantic Model to Query Spatial-Temporal Data', *The 6th International Workshop on Information Fusion and Geographic Information Systems: Environmental and Urban Challenges*, St. Petersburg, viewed 15 April, 2013, <http://hal.archivesouvertes.fr/docs/00/77/94/63/PDF/ifgis_A_Semantic_Model_to_Query_Spatial-Temporal.pdf>.

[37] Evdoxios, B, 2008, 'TOQL: Querying temporal information in ontologies', viewed 1 April 2013, <www.intelligence.tuc.gr/lib/downloadfile.php?id=293>.

[38] Terenziani, P, German, E & Shahar, Y, 2008, 'The Temporal Aspects of Clinical Guidelines' in Teije, AT, Miksch, S & Lucas, P (eds), *Computer-based Medical Guidelines and Protocols: A Primer and Current Trends*, IOS Press, Amsterdam, pp. 81-100.

[39] Terenziani P, Mastromonaco F, Molino G & Torchio M, 2000, 'Executing Clinical Guidelines: Temporal Issues', *Proc AMIA Symp. 2000*, pp. 848–852, viewed 24 April 2013, <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2244024/>.

[40] Shahar, Y, Miksch, S & Johnson, P, 1997, 'A Task-Specific Ontology for the Application and Critiquing of Time-Oriented Clinical Guidelines', *Lecture Notes in Artificial Intelligence*, vol. 1211, pp. 51-61, viewed 15 December 2012, USYD databases and electronic resources, SpringerLink.

[41] Weng C, Kahn M, & Gennari J 2000, 'Temporal Knowledge Representation for Scheduling Tasks in Clinical Trial Protocols', *Proc AMIA Symp. 2000*, pp. 848-52, viewed 5 May 2013, <http://www.ncbi.nlm.nih.gov/pubmed/11080004>.

[42] Loganantharaj, R & Giambrone, S, 1995, 'Probabilistic Approach for Representing and Reasoning with Repetitive Events', *Proc. second International Workshop on Temporal Representation and Reasoning (TIME'95)*, pp. 26-30, viewed 15 March 2013, CiteSeerX database.

[43] QUAIC, 2010, *Intensive Care Unit Empirical Antimicrobial Treatment Guidelines*, viewed 10 February 2013, <http://intensivecare.hsnet.nsw.gov.au/state-wide-guidelines>.

[44] IHTSDO, 2012, *SNOMED CT Technical Implementation Guide January 2012 International Release*, viewed 12 June 2013, <http://ihtsdo.org/fileadmin/user_upload/doc/>.

[45] Ahmadian, L, F.De Keizer, N & Cornet, R, 2009, 'The Use of SNOMED CT for Representing Concepts Used in Preoperative Guidelines' in K.-P. Adlassnig et al. (eds), *Medical Informatics in a United and Healthy Europe*, IOS Press, Amsterdam, vol. 150, pp. 658 – 662.

[46] Hitzler, P, Krötzsch, M & Rudolph, S, 2009, *Knowledge, Representation for the Semantic Web Part II: Rules for OWL*, viewed 22 March 2013, <www.semantic-web-book.org/w/images/5/5e/KI09-OWL-Rules-2.pdf>.

[47] Dean, N, 2003, *Logic and Language*, Palgrave Macmillan, New York, pp. 75-82.

[48] Oracle, 2013, *RDF Semantic Graph Developer's Guide 12c Release 1 (12.1)*, viewed 15 June 2013, < http://docs.oracle.com/cd/E16655_01/appdev.121/e17895/toc.htm >

[49] Zelenitsky, S, Iacovides, H, Harding, G & Ariano, R, 2004, 'Effect of antibiotic sequence on combination regimens against Pseudomonas aeruginosa in a multiple-dose, in vitro infection model', *Diagnostic Microbiology and Infectious Disease* vol.49, pp. 67–70, viewed 11 November 2013, USYD databases and electronic resources, ScienceDirect, DOI: 10.1016/j.diagmicrobio.2003.10.016.

[50] Allen, JF, 1983, 'Maintaining Knowledge about Temporal Intervals', *Communications of the ACM*, vol. 26, pp.832-843, viewed 25 January 2013, USYD databases and electronic resources, ACM Digital Library, DOI: 10.1145/182.358434.

[51] Nebel, B & Bürckert, HJ, 1995, 'Reasoning about Temporal Relations: A Maximal Tractable Subclass of Allen's Interval Algebra', *Journal of the ACM*, vol. 42(1), pp. 43–66, viewed 18 April 2013, USYD databases and electronic resources, ACM Digital Library, DOI: 10.1145/200836.200848.

[52] Alspaugh, TA, 2005, Software Support for Calculations in Allen's Interval Algebra, *UCI-ISR-05-02*, viewed 10 May 2013, CiteSeerX database.

[53] Saulnier, T & Trudel, A, 2005, 'Generating, Storing and Using the Complete Composition Table in Allen's Temporal Interval Algebra', *Proceeding (481) Artificial Intelligence and Soft Computing*, viewed 13 May 2013, < http://pheasant.acadiau.ca/Publications/481-131.pdf>.

[54] Krokhin, A, Jeavons, P & Jonsson, P, 2003, 'Reasoning about temporal relations: The Tractable Subalgebras of Allen's Interval Algebra', *Journal of the ACM (JACM)*, Volume 50(5), pp. 591-640, viewed 06 June 2013, USYD databases and electronic resources, ACM Digital Library, DOI: 10.1145/876638.876639.

[55] Terenziani, P, 2006, *Reasoning about Time*, viewed 10 July 2013, <http://www.di.unito.it/~terenz/PUBBLICAZIONI/B1.pdf>.

[56] Van Beek, P, 1990, *Exact and Approximate Reasoning about Qualitative Temporal Relations*, viewed 20 June 2013, CiteSeerX database.

[57] Qi, G & Harth, A, 2012, 'Reasoning with Networked Ontologies', in Suárez-Figueroa, MC, Gómez-Pérez, A, Motta, E & Gangemi, A (eds), *Ontology Engineering in a Networked World*, Springer, Heidelberg, pp. 366-367, viewed 22 September 2013, USYD databases and electronic resources, Springer Link, DOI: 10.1007/978-3-642-24794-1_17.

[58] Parsia, B, Sirin, E & Kalyanpur, A, 2005, Debugging OWL ontologies, *Proceedings of the 14th international conference on World Wide Web*, pp. 633-640, viewed 05 October 2013, USYD databases and electronic resources, ACM Digital Library, DOI: 10.1145/1060745.1060837.