S*D*TDM$^{n}_{0}$: A Multidimensional Distributed Data Mining Framework Supporting Time Series Data Analysis for Critical Care Research

by

Agam Dhanoa

A Thesis Submitted in Partial Fulfillment
Of the Requirements for the Degree of

Master of Health Science

in

The Faculty of Health Sciences

Program

University of Ontario Institute of Technology

April 2011

Statement of Authentication

The work presented in this thesis is, to the best of my knowledge and belief, original except as acknowledged in the text. I hereby declare that I have not submitted this material, either in full or in part, for a degree at this or any other institution.

....

Agam Dhanoa

Table of Contents

List of Figures

List of Tables

Abstract

Premature birth is one of the major perinatal health issues across the world. In 2007, the estimated Canadian preterm birth rate was 8.1% (CIHI, 2009). Recent research has shown that conditions, such as nosocomial infections or apnoeas, exhibit certain variations in the baby's physiological parameters which can indicate the onset of the event before it can be detected by physicians and nurses. Neonatal Intensive Care Units are some of the highest information producing areas in hospitals. The multidimensional and distributed nature of the data further adds another layer of complexity as physiological changes can occur in one data stream or can be cross-correlated between several streams. With the collection and storage of electronic data becoming a global trend, there is an opportunity to analyse the collected data in order to extract meaningful information and improve healthcare. The aforementioned properties of the data motivate the need for a framework that supports analysis and trend detection in a multidimensional and distributed environment.

Keywords: Distributed Data Mining, Temporal Abstraction, Relative Alignment, Time Series Data Analysis, NICU, Critical Care, Clinical Decision Support, Multidimensional Distributed Framework.

Acknowledgements

This dissertation would not have been possible without the guidance and the help of several individuals who in one way or another contributed and extended their valuable assistance in the preparation and completion of this study.

First and foremost, I offer my sincerest gratitude to my supervisor, Dr. Carolyn McGregor, whose sincerity and encouragement I will never forget. Dr. McGregor has been my inspiration as I hurdle all the obstacles in the completion this research work.

I also wish to thank my co-supervisors; Dr. Christina Catley and Dr. Andrew James who really helped me shape and perfect my work with their expert opinions and constructive feedback.

I would also like to thank my colleagues and friends for their consultations and moral support throughout this journey.

Finally, I thank my family for supporting me throughout all my studies and for their understanding & guidance in helping me shape my future.

We are all inventors, each sailing out on a voyage of discovery, guided each by a private chart, of which there is no duplicate. The world is all gates, all opportunities. - Ralph Waldo Emerson.

1. Chapter 1 - Introduction

Preterm birth, which occurs in 11 percent of all pregnancies in North America, is responsible for the majority of neonatal deaths (Khashan et al. 2008). The rate of premature birth has increased by 36 percent since the early 1980s (Martin et. al. 2008). Globally, premature birth and its associated complications are responsible for the mortality of one million premature babies every year (Beck et. al. 2009). In Canada, in the year 2007, the estimated Canadian preterm birth rate was 8.1% (CIHI, 2009). It is not uncommon for premature babies to spend three to four months in the Neonatal Intensive Care Unit (NICU) and suffer from a number of different conditions during their stay. Recent research is showing that these conditions, such as nosocomial infections (hospital-acquired infections), seizures and apnoeas, appear to exhibit certain early variations in the baby's physiological parameters which have the potential to be new pathophysiological markers for condition onset. Sepsis is a common nosocomial infection that affects these babies and it has been shown to exhibit changes in physiological data before the condition can be diagnosed through blood cultures (Griffin, Lake 2007). These changes require high frequency analysis of the physiological stream and are frequently not detected by physicians and nurses. Since indicative readings are mostly recorded on paper every 30 or 60 minutes by nurses, the physiological changes can often go unnoticed until the illness manifests itself fully.

Intensive Care Units (ICUs), specifically NICUs, are some of the highest information producing areas in hospitals because of the highly advanced patient monitoring equipment present in these facilities, which often output 1024 readings of waveform data every second (Stacey, 2007). The current problem is not a lack of data, but an inability to use this data for early detection of problems and intelligent decision making. There is a need for systems aimed at clinical management to help analyze complex multidimensional data produced by the monitoring devices connected to the babies. We also need clinical research systems and frameworks to facilitate retrospective analysis on stored historical physiological patient data to enable the discovery of previously unknown trends and patterns that may indicate the onset of some condition. This knowledge can then be applied to new patient cases. As mentioned, medical monitoring equipment produces large amounts of data, which makes analyzing this data manually impossible. Another level of complexity is introduced by the multidimensional nature of the data which means that there are now multiple streams of information that are interlinked and can change simultaneously, further complicating the analysis. The data is considered distributed because every NICU has its own method of data collection which can vary from one site to another due to differences in equipment and data output. In addition, these patients can be part of multicentre studies which are controlled studies executed by several cooperating institutions; leading to the possibility of a highly distributed data set.

Faced with an exponential amount of data, many organizations are turning to data mining to translate data to information and subsequent knowledge. Distributed data mining (DDM) refers to the mining of distributed data sets which are often stored in local databases and hosted by local computers connected through a network (Yongjian, 2001). Due to the advances made in computing and communication over wired and wireless networks, we can now find many distributed computing environments like the internet, intranets and local area networks. It is also important to note that many of these environments have different distributed sources of capacious data, the analysis of which requires data mining technology specific to distributed applications. Medical data is often distributed due to concerns of security, privacy and confidentiality of patient information; this is why it is likely to maintain its distributed nature in the future. The Canada Health Infoway (CHI) is an independent not-for-profit corporation created by Canada's First Ministers to foster and accelerate the development and adoption of electronic health record (EHR) systems with compatible standards and communications technologies. CHI's model currently does not provide a central solution to locate the physiological streams. In distributed data mining, the data mining occurs both at a local level and a central level. At the global level, local data mining results are combined to discover global patterns or themes present in the data.

Through the literature, DDM is also often referenced with parallel data mining which is when data mining tools are implemented on high-performance parallel computers. Both techniques aim towards improving the performance of traditional data mining systems but what sets them apart is their system architectures. DDM's main objective is to execute data mining operations based on the type and availability of the distributed resources (Park, & Kargupta, 2001). DDM techniques can be used to perform clinical research using physiological data streams in order to evaluate whether a condition onset prediction is apparent from the physiological stream behaviours prior to traditional clinical diagnosis. What distinguishes DDM from Centralized Data Mining (CDM) is the fact that a DDM system can choose how to manage and analyze data, based on the properties of computing, storage and communication capabilities, either centrally or locally in the distributed locations. In contrast, the CDM system always relies on the collection of data in a single central location before any analysis can be performed, which is not practical when data is being streamed and is arriving at a very high rate. The reason this is not practical is because the collection and storage of information centrally takes away from the opportunity to analyse streaming data as it is produced. It is also important to note that DDM systems can learn and derive models based on distributed data which means that the privacy of raw patient information is maintained because patient data does not have to leave the hospital setting/distributed location. This can prove to be very beneficial in cross-country or cross continental studies as it allows for the sharing of clinical applications without violating hospital ethics or the requirements for patient data, security and privacy. Thus, a distributed approach to analyze data is more scalable and practical especially when it is applied to data coming from a number of distinct data sites. Chapter 2 reviews major DDM algorithms and systems found in the literature.

1.1 Research Motivation

As demonstrated in chapter 2, there is an absence of flexible and distributed multidimensional approaches to data mining of time series data. Monitoring systems currently used in the NICUs are not capable of monitoring cross correlated data streams but the possibility of such a system is discussed by McGregor and Stacey (2007). These monitoring systems often have very limited on-board memory, mostly in the form of rolling memory which persists anywhere between 24 to 72 hours. The data being output from them has the potential to be streamed and stored beyond the NICU environment for higher levels of analysis (Foster and McGregor, 2006). The cost of electronic storage is decreasing rapidly and the ability to collect and store temporal data through real-time clinical monitoring has emerged as an open research area.

With the collection and storage of electronic data becoming a global trend, there is an opportunity to analyse the collected data in order to extract meaningful information and improve healthcare (Moskovitch et. al. 2007). There is mounting evidence that is being uncovered by clinical research suggesting changes in physiological stream behaviours prior to the diagnosis of certain conditions such as neonatal sepsis and apnoea (Stacey et. al. 2007) (Catley et. al. 2010). The recent momentum in such research has prompted hospitals across the globe to partake in multicenter studies which can allow for the cross site analysis of the same physiological data streams to see if they would be indicative of the same events at different hospitals. This brings the element of data distribution into context as physiological data being collected from monitoring devices may differ in format and frequency for each facility. The differences may also occur due to the physiological monitoring devices being different at each facility. As will be seen in chapter 2, the lack of a multidimensional distributed temporal data mining framework drives the motivation behind this research.

1.2 Research Aims and Objectives

The lack of a multidimensional distributed temporal data mining framework which can support multicenter studies formed the research motivation for the first, second and fourth hypothesis. The need for a structure to run the temporal abstractions and relative alignments motivated the third hypothesis. Therefore the primary hypotheses for this research are:

1. A multidimensional distributed data mining framework can be defined for time series data research for the discovery of trends and patterns prior to a given clinical event.
2. The framework will utilize elements of data fusion and agent-based analysis so that it will work with relational databases and large scale data mining applications.
3. A set of data mining tools can be applied for temporal abstraction, relative alignment and cluster analysis in a distributed manner to support multiple research studies.

4. The framework can be applied in a broad neonatal context addressing issues of data privacy and confidentiality and being deployable as part of multicenter studies while maintaining data integrity at each participating site.

1.3 Research Method

This research follows a constructive methodology which is a commonly used computer science research method but has also been utilized in information systems and medical domains (Martin & Maojo, 2002). The construct can be a new theory, algorithm, model, software, or a framework which can allow us to draw theoretical conclusions. Constructive research will allow us to develop a distributed data mining framework which can contribute to the discovery of trends and patterns from medical stream data. The term construct is used in this context to refer to the new contribution being developed. Figure 2 (below) outlines the common elements found in constructive research.

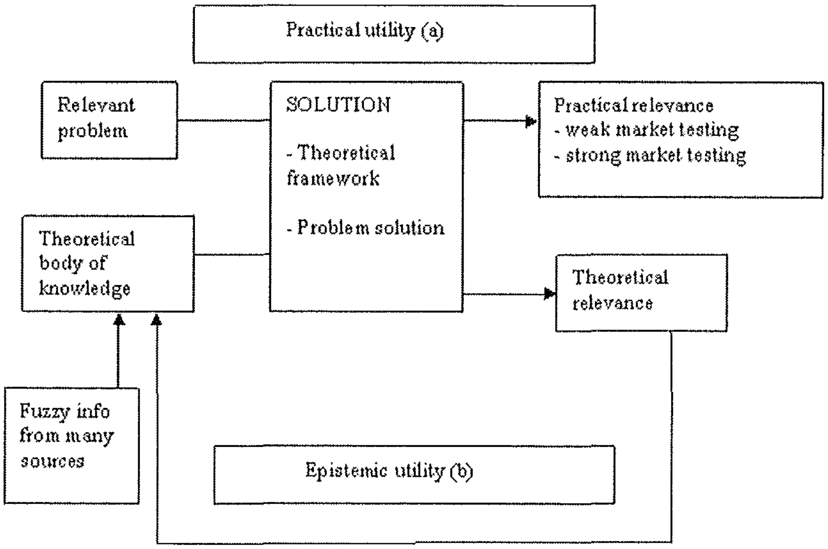


Figure .0 - Outline for Constructive Research (Constantinescu, 2005)

In this research, the problem that will be addressed is the need to use distributed multidimensional data to perform temporal abstractions and relative alignments which can help identify trends and patterns being exhibited in data coming from multiple sources. This can aid in the early detection of the onset of patients of interest for specific conditions in the NICU.

In order to fully understand the topic and problem at hand, a thorough review of the literature was completed in the areas of distributed data mining, temporal abstraction and knowledge discovery in databases (KDD). This helped in designing a framework that can handle multidimensional data coming from distributed sources allowing data mining for relevant information and providing the necessary alerts when required.

1.4 Contribution to Knowledge

There are several unique challenges to support clinical research for critical care health informatics in a distributed setting. By investigating the scholarly knowledge domain, we can obtain an understanding of the issues and gain insight into the specifics of knowledge discovery and data mining in the medical space. By applying this research within the context of neonatal care we can demonstrate a real world solution that can be applied in the NICU setting as well as evaluate its applicability in other areas. The areas of research contribution to knowledge resulting from this thesis are:

- Extensions to the existing Service based Multi-Dimensional Temporal Data Mining ($STDM^n_0$) framework to support a distributed multidimensional environment.
- Design of a framework to:
    - Enable the distribution of Temporal Rules in a multi-dimensional environment
    - Support the multi-dimensional distribution of Relative Rules

- Support distribution of Rule Base data which can be deployed for real time analysis

1.5 Thesis Overview

Chapter 2 presents a literature review of the areas of influence for this thesis, mainly DDM, Distributed Data Mining of Time Series Data, Temporal Abstraction and Distributed Temporal Abstraction. The chapter explores these areas in their application to medical systems in order to highlight the open health informatics research areas leading to the development of the research hypotheses addressed by the techniques proposed in this research. Chapter 3 discusses the physical context by describing the NICU environment which provides the setting for the Service Based Multi-Dimensional Distributed Temporal Data Mining (S***D***TDM$^n{}_0$) framework designed and presented in this thesis. Chapter 4 begins by presenting the existing STDM$^n{}_0$ architecture, presenting how both static and streaming data, as well as temporal and relative temporal data are used and integrated in the STDM$^n{}_0$ framework. Chapter 5 presents the revised S***D***TDM$^n{}_0$ framework and highlights areas of distribution. Chapter 6 demonstrates how the S***D***TDM$^n{}_0$ framework can be used for conducting clinical research within a distributed NICU context. Chapter 7 concludes the thesis, summarising the research contributions and providing directions for future research.

2. Chapter 2 – Literature Review

2.1 Introduction

The main motivation for this research is the lack of distributed multidimensional approaches to data mining of time series data. The intensive care environment, where observations of the patient's condition is supported through the provision of several physiological time series data streams via medical monitors, presents an opportunity for discovering new knowledge that may exist in patient data indicative of the onset of specific conditions. A distributed multi-dimensional framework can allow for the possibility of running abstractions across multiple locations simultaneously while keeping data consistent across sites. In this chapter, the area of DDM is introduced followed by a review of DDM in relation to time series data. Next the area of temporal abstraction is reviewed followed by a detailed look into literature relating to distributed temporal abstractions.

2.2 Distributed Data Mining (DDM)

Distributed Data Mining (DDM) involves the use of distributed data analysis algorithms as well as distributed systems. Throughout the literature, the use of the Multi-Agent System (MAS) has been a common theme for many DDM systems (Ferber, 1999). The MAS has been developed from the Distributed Artificial Intelligence (DAI) which focuses on artificial intelligence based search, learning, planning and problem solving techniques for distributed environments (Ferber, 1999). Existing literature on multi-agent systems and learning do not address the issues of large scale distributed data analysis. The MAS focuses on learning control knowledge and adaptive behaviour (Byrne & Edwards, 1995).

The concept of data fusion is also an important finding in the literature and refers to the seamless integration of data from disparate sources (Park, & Kargupta, 2001). Within the context of data fusion, the distributed approach of multi sensor data fusion is closely linked to our scenario. This approach discusses sensors that make a local decision based on the raw data and then combine all of the individual local decisions at a fusion center in order to produce a global decision. This not only maintains the privacy of the raw data, but also helps maximize the probability of determining the optimum local and global decision rules which in turn help make signal detection much more accurate. Making these decisions also relies on hypothesis testing techniques which are often done using the the Bayesian criteria (Hoballah & Varshney, 1989) or the Neyman-Pearson criteria (Vishvanathan & Varshney, 1997).

Determining how the data is distributed is the starting point in developing a distributed data mining solution (Park, & Kargupta, 2001). Throughout the literature on DDM, two assumptions are commonly made when it comes to the concept of how data is distributed across multiple sites: 1) either the data is distributed homogeneously, that is partitioned horizontally or 2) the data is distributed heterogeneously, that is partitioned vertically. This relates back to the relational database in which the database schema provides the information on the relations it stores. By identifying the different schema from different tables, we are able to identify their shared dependencies which in turn help determine the type of data mining algorithm best suited for it. The majority of existing DDM algorithms assume that the data is distributed homogeneously across different sites meaning that each distributed site contains the same set of attributes. The heterogeneous scenario assumes that each site contains a collection of columns and do not have the same set of attributes and each tuple is assumed to contain a unique key column that links corresponding rows across the tables.

As discussed, a distributed approach to data analysis is more scalable. Agent based data mining is one way to perform scalable mining on large data sets that contain distributed data. The agent based DDM systems use one or multiple agents on each data site. These agents are responsible for the analysis of local data and for communicating with other agents during the mining phase. Once the local data has been mined for knowledge or patterns, it is pooled together in a global cloud of synthesized knowledge. In order to keep the local agents at optimal mining performance and keep control over the resources, agent based systems require a "supervisor agent" often called a facilitator that controls the behaviour of each local agent. BODHI and JAM (Java Agents for Meta-learning) are two DDM architectures that follow the supervisor agent concept, are able to work with relational databases, support large scale data mining applications, and can be implemented using Java which also makes them platform independent; all of which make them well suited for our environment. The JAM system not only provides distributed data mining capabilities, it also allows a user to monitor and visualize the various learning agents and derives models in real time (Stolfo et al. 1997).

2.3 Distributed Data Mining of Time Series Data

Abe & Yamaguchi discuss an integrated time series data mining environment (Abe & Yamaguchi 2005). The design integrates time series pattern extraction methods, rule induction methods and rule evaluation methods with active human interaction. The authors suggest that time series rules can prove to be an important form of medical evidence but it is often difficult to find such evidence systematically. This limitation motivated their development of a time series data mining environment which applies data mining techniques to systematically discover medical evidence. The cooperation of data miners, system developers and domain experts are also key factors in the success of such an environment. The authors present a hepatitis related case study to help identify procedures needed to execute time series data mining cooperatively with active human system interaction. In the study, interferon (IFN) treatment results data are taken as the target for the representation as if-then classification rules. Two key phases are identified: firstly, rousing a new hypothesis in the expert is said to be key in gathering interest from the domain experts as this adds in a level of involvement for the expert. In this first phase, patterns extracted from historic data are presented to the physician to help identify the distinguishing patterns from their perspective. The next step involves the validation of identified patterns with each patient sequence to determine pattern 0: the typical course; pattern 1: the least reaction; and pattern 2: the adverse reaction to IFN treatment. The second phase involves ensuring the expert's hypothesis which involves the extraction of patterns for 40 weeks as the observation period. Next, patterns are joined as attributes of data sets from which if-then rules are induced within the data set. Then, patterns and rules are presented to the physician for evaluation using a Graphical User Interface (GUI). The authors make an important point about present time series data analysis techniques which is that they mainly utilize statistical methods like the Autoregressive Integrated Moving Average (ARIMA) which are suited for well-formed data. It is also noted that present signal processing methods like Fourier transform, wavelet and fractal analysis are used to analyze well-formed time series data but the problem that exists for medical data is that it is mostly ill-formed, meaning that they can include data such as clinical test data, purchase and financial data. To combat this issue, the authors suggest the use of Dynamic Time Warping (DTW) which utilizes time series clustering with multi-scale matching of data. The authors conclude their work by identifying key procedures in time series data mining frameworks, which include: procedures for pattern extraction i.e. data pre-processing, rule induction i.e. mining and the evaluation of rules with a visualized rule i.e. post-processing of mined results.

2.4 Temporal Abstraction

Clinical time series data is often collected in large volumes but very little is done to analyze, interpret and extract from these data sets. Temporal Abstraction (TA) is an important piece in the development of clinically relevant evidence based support systems. TA adds qualitative information to generally quantitative data which allows us to identify the patterns or trends present in the data set. This is important because almost all clinical data has an associated temporal dimension (Dolin, 1995); for example, most diseases have an onset or a set duration, studying which can help in early detection of their onset or progression. Thus, automated systems that work with clinical data must be able to reason and cope with this type of input which is often called temporal reasoning (O' Connor et. al. 2002).

One of the core steps in forming temporal reasoning is the creation of high level temporally extended concepts from raw time-stamped data which is often referred to as temporal abstraction (Shahar, 1997). To further add to the complexity, clinical data can often be multi-dimensional and distributed across multiple sites. It is often seen that an increase in data frequency, distribution and dimensionality is directly proportional to the complexity of the potential trends and patterns that can be observed in clinical data (Catley et. al. 2010). Temporal Data Mining (TDM) is an emerging area of research that helps with this problem as it integrates the TA processes involved in trend and pattern detection with new knowledge gained from data mining.

Intensive Care Units (ICUs) are some of the highest information producing areas in hospitals because of the highly advanced patient monitoring equipment present in these facilities which often output 1024 readings every millisecond (McGregor and Stacey, 2007). There is a need for TA and TDM systems aimed at clinical management to help analyze complex multidimensional data produced by the monitoring devices connected to the patients and derive relationships from that data which can help in earlier diagnosis and treatment of conditions.

2.5 Distributed Temporal Abstraction

This section presents a review of the Distributed Temporal Abstraction Systems that have been developed to date. The aim was to determine how existing distributed systems are designed and to review the degree to which security, privacy and confidentiality are considered in the design of current distributed temporal abstraction systems. In addition we sought further review of the function of TA in a distributed setting.

Medical data is often distributed and stored locally with each healthcare provider due to concerns of security, privacy and confidentiality of patient information. This is why it is likely to maintain its distributed nature in the future as policies encompassing patient privacy improve. Most of the systems that were reviewed however, do not discuss much with respect to data privacy and security. The systems that have been reviewed do not discuss the possibility of a distributed storage of TA but rather assume a local data storage model with a very minimal amount of distribution i.e. mainly the distribution of the TA queries only or a distributed data collection method.

Shahar et al. (1998) present the Asgaard framework used to abstract raw monitoring data collected by NICU monitoring devices to the abstract concepts that are used in therapeutic plans. The authors explain the nature of the data that is being output from the NICU as a stream of high frequency raw information. The framework involves the high-level abstraction derived from the raw data which is then compared to predefined conditions described in the therapeutic plans. The authors also discuss the systems and languages that are currently used to abstract and store data at present. They mainly focus on the system named RÉSUMÉ which is traditionally used on low frequency data. This poses a problem because the data stream output from the NICUs monitoring devices is at a high frequency rate and may cause the RÉSUMÉ system to overlook some of the underlying patterns in the output data.

O'Connor et. al. (2002) present a Distributed Temporal Abstraction System which allows for the facilitation of knowledge-driven monitoring of clinical databases. The system, named A System for Temporal Abstraction (RASTA), is based on a component based architecture called EON which was developed by the authors as a means for building automated clinical decision support systems. The EON architecture made use of RÉSUMÉ as the knowledge-based system for performing temporal reasoning. The authors further discuss the problem with RÉSUMÉ which is the fact that it does not scale to the significantly higher data processing requirements for working with large amounts of data. Another issue is that it is a stand-alone rule-based system and does not offer real-time response rates for anything other than small single-patient data sets. RÉSUMÉ also does not allow the abstraction tasks to be distributed. Finally, RÉSUMÉ has an exponential relationship between the size of the data set it operates on and its memory and CPU requirements.

In order to address the aforementioned issues the authors propose the use of the RASTA. In many ways, RASTA is an extension to RÉSUMÉ as it incorporates many of the ideas and concepts used by the latter. The authors explain that RASTA uses a distributed algorithm that allows independent evaluation of each abstraction in an abstraction hierarchy which allows it to use separate processes for each portion of an abstraction tree for each patient. Thus working on very large data sets does not cause any problems in relation to memory or CPU consumption making this system much more efficient. The authors also describe the modularity of RASTA as it can be deployed as a single standalone process if enough resources are available or it can be distributed across multiple processes on multiple machines.

The algorithm used by RASTA for temporal abstraction involves four main data sources: (1) Domain knowledge base (2) Time stamped data (3) Contextual data and (4) Case identifiers. RASTA also draws some subtasks in its temporal abstraction algorithm from Shahar's knowledge-based temporal-abstraction problem-solving method (Shahar and Musen, 1993). These include (1) Context Restriction (2) Vertical Temporal Inference (3) Horizontal Temporal Inference and (4) Temporal Interpolation. All of the aforementioned subtasks work well for RÉSUMÉ but are again not ideal for a distributed and high volume data set. The authors further discuss how horizontal temporal inferences and temporal interpolations can be very expensive computationally. An increase in raw time stamped data points means the response time when performing TA will also increase significantly. The response time for these abstractions can be acceptable for single patient data but when dealing with multi patient data the response rate can become unacceptably long. The authors address this issue in RASTA by building a TA algorithm that is parallelizable and distributable (O' Connor et. al. 2002). Details on the algorithm can be found on Page 3 of the paper.

Finally, the implementation of RASTA has also been designed in a way that it stays modular and extensible. RASTA is written in Java and uses CORBA (Vinoski, 2002) as its inter-process communication mechanism. RASTA also uses the XML format for the data that is exchanged between processes. All knowledge bases used by RASTA are written using Protégé-2000 (O Connor et. al. 2002) which is a knowledge base authoring environment with the main benefit being that it provides automated assistance in the acquisition of abstraction knowledge from domain experts (O Connor et. al. 2002). It is important to note though that there is no standard way for knowledge collection that is discussed by the authors.

Boaz and Shahar, (2005) present a distributed temporal-abstraction mediator for medical databases known as Idan. The authors claim the need for an integration of data and knowledge in clinical practice. Most stored data include a time stamp in which the particular datum was valid and the authors outline the need for a system that can automatically create abstractions of time oriented clinical data and be able to answer queries about the abstractions (Boaz and Shahar, 2005). The key to the success of such a system, according to Boaz and Shahar, is the intelligent integration of knowledge sources, data sources and computational services. They emphasize the fact that any distributed TA system must be modular and at the same time support knowledge and data sharing. The authors also note that data, knowledge and computational services might be integrated in multiple configurations which demand that the TA architecture be distributed and possibly accessible via the Internet. Boaz and Shahar further emphasize that the system should exploit domain specific knowledge and should be able to support several modes of interaction by various applications that use its services (Boaz and Shahar, 2005).

One of the main parts of the Idan architecture is the temporal abstraction mediation. Temporal reasoning and temporal data maintenance i.e. storage, query and retrieval of time-oriented data, must often be performed at the same time in order to support clinical needs (Boaz and Shahar, 2005). A temporal database mediator "mediates" time oriented queries from decision support applications to patient databases. It acts as an intermediate layer of processing between client applications and databases and is not reliant on a particular application or a particular database. This type of temporal abstraction mediator has been discussed by Nguyen et. al. in the Tzolkin system (Nguyen et. al., 1999). According to the authors, implementation of a temporal abstraction mediator has many advantages as it can be task specific and domain independent but it also must use standard controlled medical vocabularies to support sharing of data and knowledge as much as possible.

Thus the Idan architecture, being a modular distributed TA mediator fully implements the temporal-abstraction mediation approach discussed earlier. The main integration points in Idan are (1) time oriented data sources (2) domain specific knowledge sources (3) vocabulary servers (4) a computational process specific to the task of abstraction of time oriented data using domain specific knowledge and (5) a controller for the integration of all services. All of these points have been shown diagrammatically by the authors in the figure below.

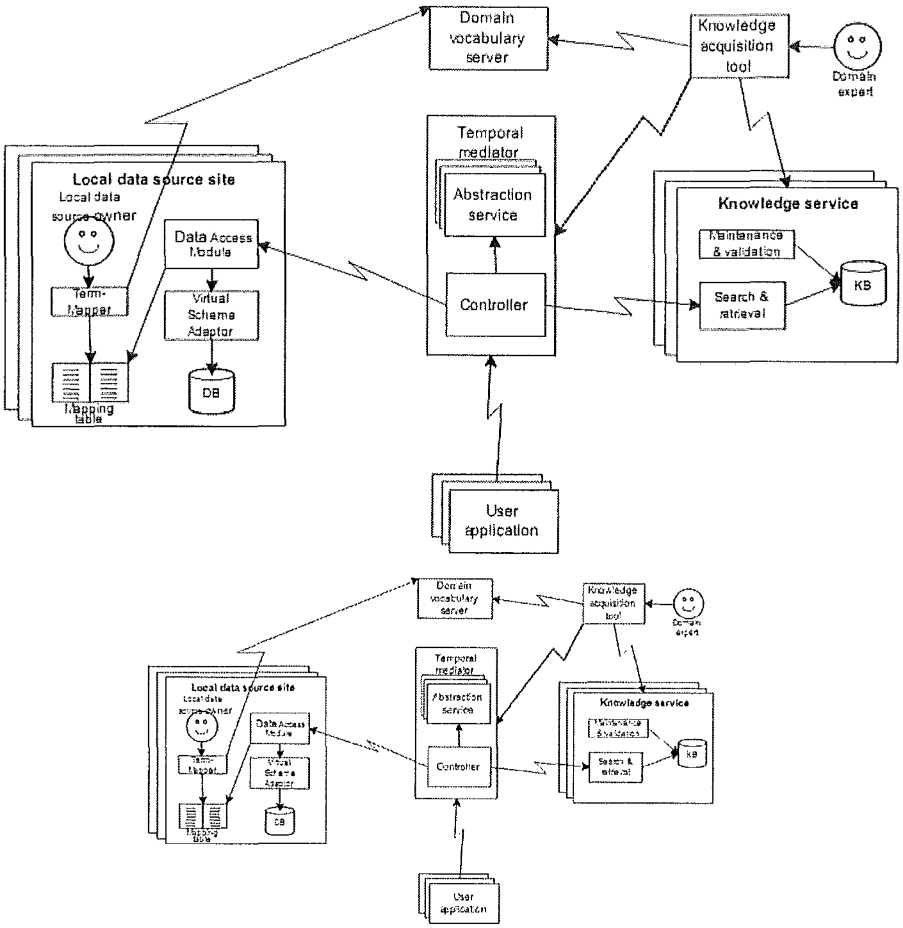


Figure 2.2 - The Idan Architechture. (Boaz and Shahar, 2005)

Idan is able to answer abstract, time-oriented queries by adequately handling the queries to the various key modules in a distributed system (Boaz and Shahar, 2005). Where it differs from RASTA is that it is capable of handling temporal constraints in a uniform way between the system level and the interface level.

Stonebaker et. al. (1996) discusses an architecture for distributed data called Mariposa. One of the main objectives of this architecture is to unify disparate approaches of distributed database management systems (DBMS). Mariposa works by distributing data over a number of sites that can be connected via LAN or WAN. In addition to this, Mariposa requires each site to have a storage device and in case multiple storage devices are connected, then Mariposa considers them as a secondary site.

The authors describe the Mariposa database as one consisting of instances of objects in named classes each of which contain a collection of attributes of the specific data types. The Mariposa database uses a fragment storage system in which each class is divided into a collection of fragments. The authors also state that these fragments can be shared across sites as they do not have a specific home and can move freely within a network. The Mariposa system also organizes these fragments based on usage i.e. if a fragment is being accessed frequently, the system will allot that fragment more resources.

User control for the various sites in the Mariposa Architecture is also locally controlled by a database administrator. Having such local control helps database administrators specify local rules for that storage site. For example, if storage space at one site reaches maximum capacity, then the system references the storage rule set by the administrator which can tell it where to move a specific fragment of data or what to delete.

The authors also outline a rule processing subsystem that is part of the Mariposa architecture. Every Mariposa site runs an instance of the rule processor which watches for events of interest, the criteria or policies for which can be pre-determined and programmed into the system. Conventional systems, according to the authors, make changing these “policies” quite difficult as they are hard coded into the system. Mariposa, on the other hand, allows these policies to be changed dynamically at any site across the network.

Table 2.1 summarizes the findings detailed above and highlights areas that are largely unaddressed in current studies.

Systems	Environment	Freq.	Multiple Streams?	Real-time?	Privacy	Dist. Abst.	Locations	TA Deployment
Asgaard | Shahar et al. 1998	NICU – Abstractions for therapeutic plans	Can only handle Low frequency data – Based on RÉSUMÉ	No	No	No	No	Single Site	Yes
RASTA | O'Connor et. al. 2002	Knowledge-driven monitoring of clinical databases	Low frequency data – Uses EON - Based on RÉSUMÉ	Parallelizable and Distributable	No	No	Based on abstraction hierarchy (an abstraction tree for each patient) – focus on memory	Single Site	No
Idan | Boaz and Shahar, 2005	Integration of data and knowledge in clinical practice	Low frequency retrospective data + data form domain experts	No	No	No	TA mediator for time oriented data sources + domain specific knowledge sources + vocabulary servers	Single Site	Yes
Mariposa | Stonebaker et. al. 1996	unify disparate approaches of distributed database management systems – IT Environment	Not applicable - distributing data over a number of sites that can be connected (WAN or LAN)	None – Focus on database management not clinical data	Yes - rule processor which watches for events of interest (pre-determined and programmed)	No	None but rule processor discussed for real time analysis	Distributed	No

Table 2. - A Summary of Existing Distributed Temporal Abstraction Systems and their Shortfalls

2.6 Conclusions and Implications of Research

There are several challenges in developing a distributed data mining framework able to work in a multidimensional environment. Ability to handle varied data frequencies, considerations on data privacy and the location of where patient data exists, ability to handle real time stream data and the synchronous deployment of abstractions for data consistency are key considerations towards designing a functional framework. In order to enable the discovery of new trends and patterns that may be indicative of the onset of a condition in patients, there is a need for an integrated multidimensional distributed data mining framework.

As a result of the investigations from the literature the research hypotheses as presented in chapter 1 were determined.

3. Chapter 3 – The NICU Environment

This chapter presents an overview of the clinical environment which provides intensive care for newborn babies, often referred to as neonates, during their first 28 days of life. Neonates admitted to an intensive care unit may be seriously ill full-term babies or babies born prematurely.

The Neonatal Intensive Care Unit (NICU) is the unit of a hospital specializing in the care of premature and critically ill newborn infants. NICUs were developed in the 1950s and 1960s by paediatricians to provide better isolation from infection, better temperature support, and greater access to specialized resources and equipment (Hilberman, 1975). NICUs often deal with premature babies who require constant monitoring and care. These babies are also at a high risk of developing multiple complications during their hospitalisation so it is crucial to keep track of their condition at all times.

About 8% of babies born in Canada each year are born premature, and many of them require extra support. Speaking globally, every year one million premature babies around the world do not survive, according to the March of Dimes (Beck et. al. 2009). Graduates of the NICU have higher rates of learning disabilities, respiratory illness and can have a higher incidence of developmental and behavioural problems (Kramer et. al. 2002).

3.1 The Canadian Context

Canadian NICUs follow a highly regionalized system of neonatal care (CPS, 2006). The concept of a regionalized system for hospitals that care for newborn infants according to the level of complexity of care that is provided was first proposed in 1970 by the Canadian neonatologist Paul Swyer (CPS, 2006); this was later put into action in the year 1976 following a March of Dimes report (Stark, 2004). There are three main levels of care (CPS, 2006):

1. Level one (normal newborn care)
2. Level two (high dependency care) and
3. Level three (intensive care)

Level one, two and three NICUs are strategically located within health regions: some very small regions may not have a level three NICU. Patients in level 1 NICUs are considered to be normal newborn infants aged 34 weeks gestation or higher (CPS, 2006). The units at this level have to be equipped to evaluate healthy newborns and provide postnatal care, perform neonatal resuscitation if needed and stabilize infants until they are transferred to an appropriate higher level facility if needed. Level 2 NICUs provide care to moderately ill infants with problems that are expected to resolve soon or who are recovering after intensive care treatment. Infants in level 2 NICUs are aged 32 weeks gestation or higher. Finally, level 3 NICUs support critically ill newborn infants as well as infants that require surgical intervention. Infants assigned here generally require an intricate level of care and typically have the longest length of stay from all other levels.

3.2 The "Wired" Neonate

It is common for the neonates in the NICU to undergo numerous medical diagnoses, procedures and other treatments. All of these require constant supervision by NICU clinicians and the sophisticated equipment of the NICU comes to the clinician's aid. At any given point, a neonate may be connected to multiple devices performing both analytical and support tasks in the NICU and generating a plethora of information. Figure 3.1 shows a typical NICU bed space and highlights the range of devices that a baby can be connected to during their stay. In The Hospital for Sick Children, for example, the Phillips IntelliVue MP70 monitoring devices play an important role in collecting and displaying data such as heart rate, transcutaneous oxygen saturation (SpO_2), electrocardiogram (ECG), blood pressure, and respiration rate.

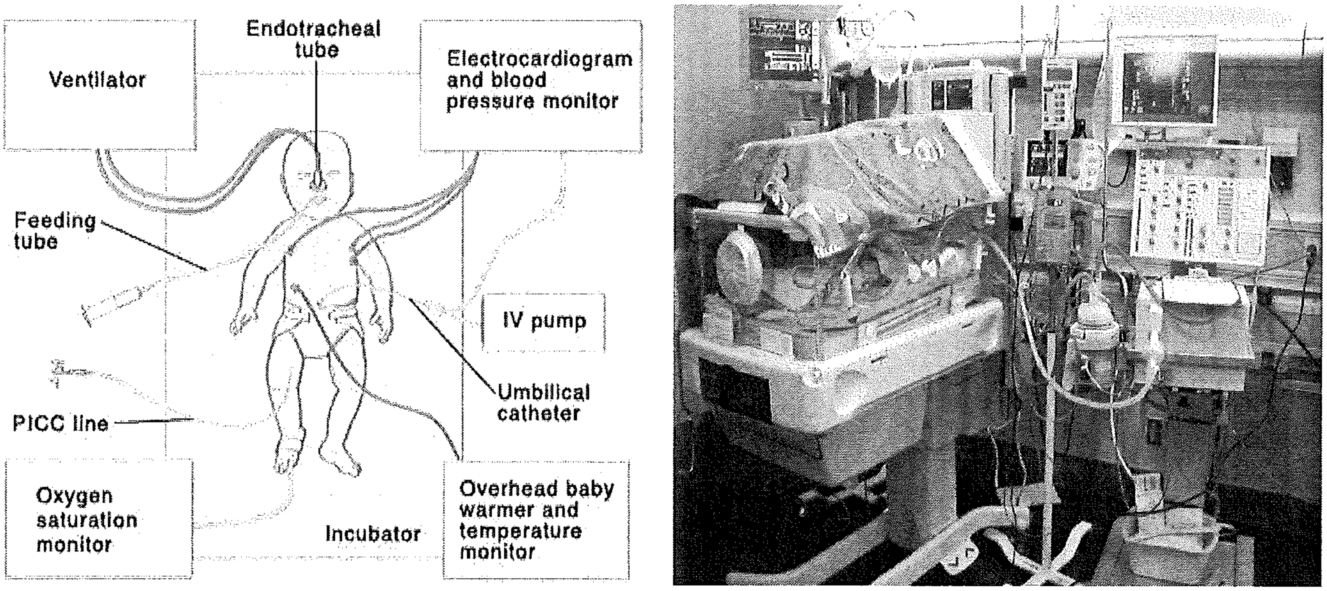

Ventilator
Endotracheal tube
Electrocardiogram and blood pressure monitor
Feeding tube
IV pump
Umbilical catheter
PICC line
Oxygen saturation monitor
Incubator
Overhead baby warmer and temperature monitor

Figure 3. - A display of the typical NICU environment

3.2.1 Technology in the NICU

The constant close supervision of NICU patients is assisted by the use of a wide variety of medical devices, some of which include (Neonatology on the Web, 2002):

1. Incubators and/or radiant warmers
2. Physiologic or cardiorespiratory monitors
3. Transcutaneous oxygen saturation monitors for pulse oximetry
4. Intravenous infusion pumps

5. Phototherapy lights
6. Mechanical ventilators

The graphic below shows a typical NICU physiological monitor which would be used by physicians, nurses and respiratory therapists for monitoring heart rate and rhythm, breathing rate and blood pressure.

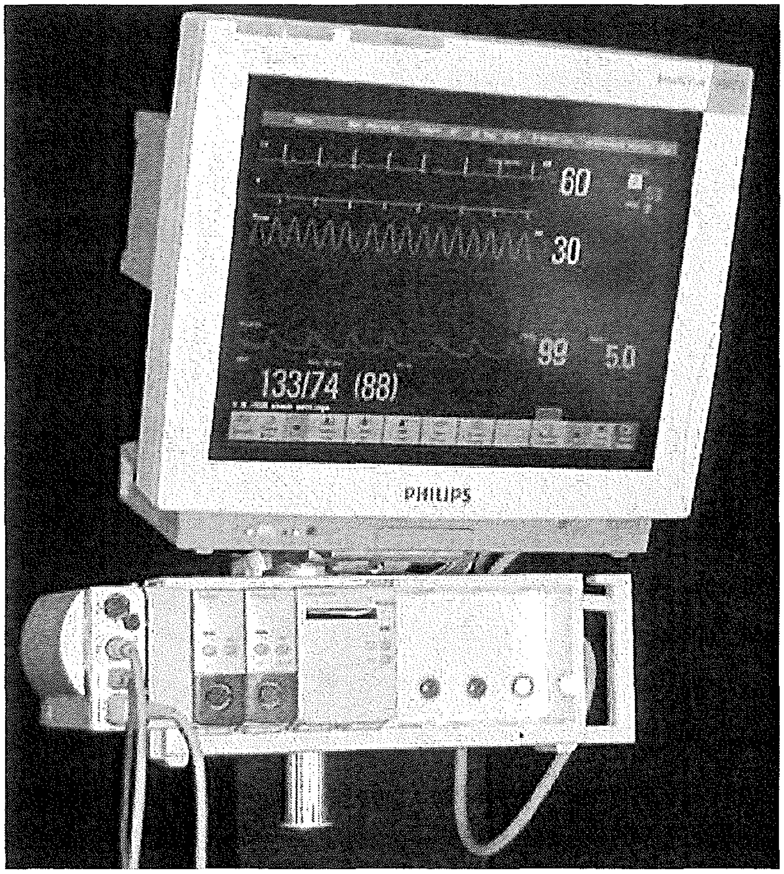


Figure 3. - A typical NICU physiological monitor

These devices often have very limited on-board memory, mostly in the form of rolling memory which lasts anywhere between 24 to 72 hours. Thus the data being output often needs to be streamed and stored beyond the NICU environment for higher levels of analysis (Foster and McGregor, 2006). A higher level of analysis is needed because in the NICU, nurses often perform routine checks on infants and record information on paper based or electronic data sheets every 30 to 60 minutes. However, it is common for critically ill neonates to have a significantly abnormal variation in the measured parameters every minute which can easily be missed in the 30 or 60 minute readings captured by nurses (McGregor and Eklund 2008).

3.4 Understanding the data rich environment

The NICU environment can often prove to be data rich yet information poor. The data intensive nature of this environment creates situations where physicians are faced with an overwhelming number of variables when caring for an infant. Miller, (1956) claims that even seasoned physicians are often unable to develop a systematic response to problems that involve more than seven variables (Miller, 1956). Data collected from the aforementioned monitoring systems can reach millions of entries in a database. Thus the data being collected provides no usable information due to the sheer volume of stored information. Data needs to be extracted and organized to become information, and a domain expert must then interpret this information before it becomes knowledge.

There are two forms of data that can be defined in the NICU environment. First, the physiological data which is collected from sensory and monitoring devices like the ones discussed in Section 3.2.1. Secondly, the clinical data which may include information on patient age, weight, paper notes or periodic readings taken by nurses. The physiological data is comprised of data streams, often acquired at varying frequencies. For example, the Phillips Component Management System (CMS) outputs the following types of data streams:

1. Numeric - a reading generated every 1024 milliseconds
2. Wave - every 32 milliseconds four data values arrive via the wave data stream (128 values every 1024 milliseconds)
3. Fast Wave - 16 values arrive every 32 milliseconds (512 values every 1024 milliseconds).

Similarly, the Phillips IntelliVue series of patient monitors can stream fast wave data values of one every 1024 milliseconds. The frequency at which data is generated may vary between different devices and manufacturers.

3.5 Existing Physiological Onset Predictors

There is mounting clinical evidence suggesting changes in physiological stream behaviours prior to the diagnosis of certain conditions. Stacey et. al. (2007) state that enabling TA to be applied across multiple patients within the NICU offers the potential of early detection of conditions such as sepsis which may exhibit early warning characteristics before being diagnosed through traditional means.

There are several conditions of interest affecting patients in the NICU. Infection is a very common cause of morbidity and an important cause of mortality for the newborn infant. Although many infants acquire their infection around the time of delivery, others acquire an infection while receiving intensive care in the NICU. These are referred to as hospital-acquired or nosocomial infections. The early diagnosis of a nosocomial infection is difficult, because the clinical signs of infection are usually subtle and nonspecific until the infection is well established (Blount et. al., 2010). These infections can occur 48 hours or more after birth and data indicates that almost 30% of infants born at 25–28 weeks gestation and more than 45% of infants born prior to 25 weeks gestation will experience a serious nosocomial infection while in the NICU

(Blount et. al., 2010). Intraventricular Hemorrhage (IVH) is another common cause of morbidity and mortality for the newborn infant. Approximately 20% of preterm infants develop an IVH. The haemorrhages occur during the first few days of life and more than 90% of the IVHs have occurred by the third day of life (Blount et. al., 2010).

Additionally, Catley et. al. (2010) discussed a framework to model and translate clinical rules to support complex real-time analysis of both synchronous physiological data and asynchronous clinical data. The authors demonstrate how a clinical rule for detecting an apneic event is modeled across multiple physiological data streams; these included a lapse in respiration rate (RR) of a neonate for greater than 15 seconds and a fall in peripheral oxygen saturation less than 85% for greater than 20 seconds combined with a heart rate of less than 100 BPM. Thus, there is increased interest and research in the early detection of the clinical decline of the patient as knowledge of early indicators of medical conditions can be made available to clinicians as soon as they are detected allowing for better patient outcomes.

3.6 NICU Clinical Research: A Distributed Problem

Patterns can be detected in the physiological data if the data from devices is captured and stored in data warehouses and is available for data mining. The main reason for storing and mining this data would be to discover previously unknown trends and patterns across various parameters and the establishment of indicators of the onset of conditions that may have an adverse effect on outcomes. Looking at this from the perspective of a distributed data environment that involves multiple hospitals across the globe, both the type and frequency at which data is being output may differ from one site to another. The differences may also occur due to the physiological monitoring devices being different at each facility. As an example, The

Hospital for Sick Children, Toronto makes use of the Philips IntelliVue MP70 series of patient monitoring devices in their NICU. The Shenzhen Maternity and Children's Hospital, Shenzhen, China makes use of the Dräger Infinity Delta XL series of monitors and the Women & Infants Hospital in Providence, Rhode Island makes use of the Spacelabs Ultraview SL series patient monitors. Not only can the format and frequency of data output differ between these devices, the frequency at which this data can be streamed to the data warehouse can also vary for each site. In addition, the three levels of NICU care can add another level of complexity and distribution as a patient graduates or is moved from one NICU to another. Thus, the need for a distributed data mining framework is quite evident when dealing with multicenter studies.

3.7 Conclusions and Implications of this Research

This chapter has introduced the NICU case study context and supported the motivation of hypothesis 4 of this thesis:

4. The framework can be applied in a broad neonatal context addressing issues of data privacy and confidentiality and being deployable as part of multicenter studies while maintaining data integrity at each participating site.

The known physiological behaviours in combination with previous non computing related clinical research form the motivation for this research. The need for a distributed data mining environment that can support multicenter studies also provides the context for the case study demonstration in chapter 6 of this thesis.

Section 3.6 of this chapter introduced the distributed problem within the context of the NICU. The intensive nature of the medical care provided to neonates in the NICU is not dissimilar to the intensive care provided in adult intensive care units; thus the applications discussed in this thesis can be extended beyond the NICU environment in the future.

4. Chapter 4 – The Existing Architecture

Chapter 2 demonstrated a current lack of frameworks to support distributed data mining environments. Heterogeneous data environments demand data mining frameworks that can normalize data in order to make them consistent across sites. This chapter presents details of the Service Based Multi-Dimensional Temporal Data Mining ($STDM^n_0$) framework, highlighting the need for its operation in a distributed setting and discussing current limitations that make distributed deployment impossible. These challenges are addressed in Chapter 5.

4.1 Components of the Existing Framework

The $STDM^n_0$ framework (Figure 4.1) focuses on reducing the gap between clinical management and clinical research (Bjering & McGregor, 2010, McGregor C. P., 2010); allowing for the effective use of the large volumes of data being collected from medical monitoring devices and stored in medical databases. The framework comprises three main components or layers i.e. the multi-agent system which is driving the framework, the extended CRISP-DM model layer which defines the data mining tasks and the $STDM^n_0$ framework task layer. In addition, there is also a layer for web services, active rules ontology, and data management. The n and 0 in the $STDM^n_0$ framework represent the data mining extensions incorporating null hypothesis and the dimensionality. The following sections explain the details of the existing framework which leads into a discussion of the extensions that will be made to the framework to enable its use in a distributed setting.

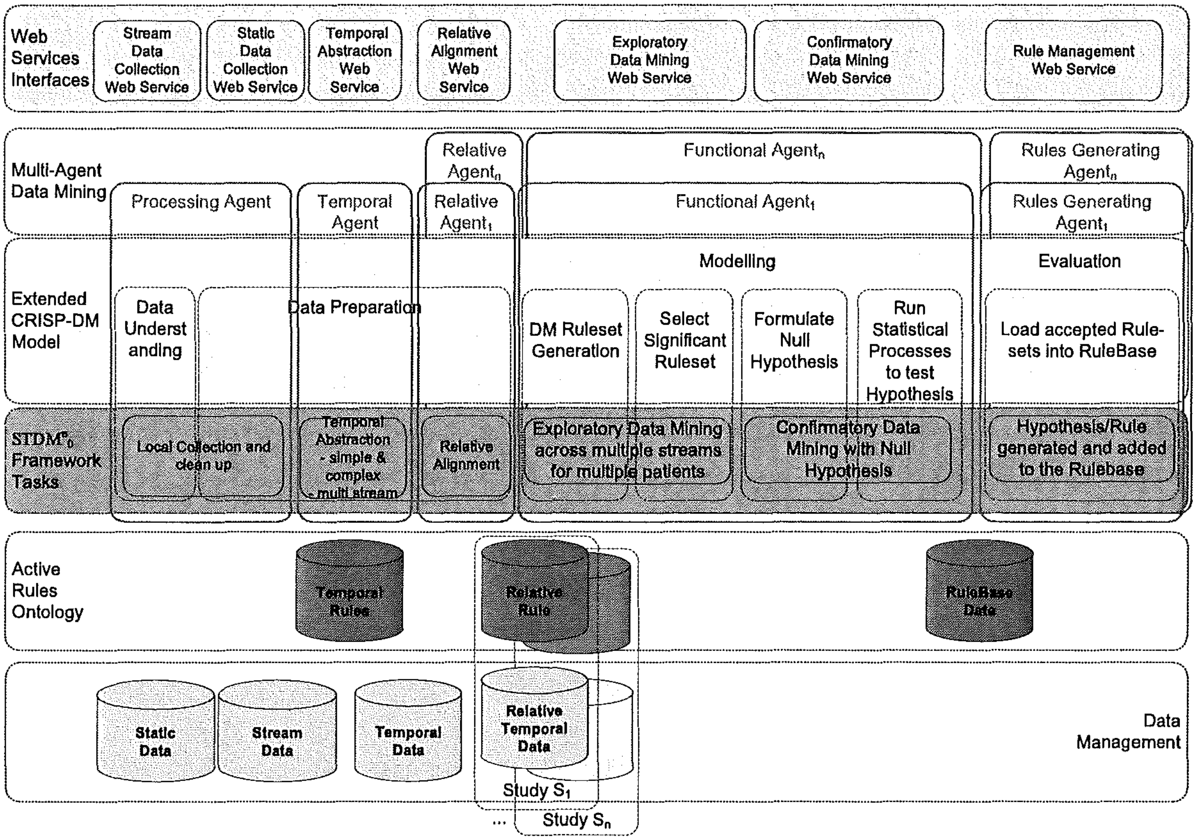

Web Services Interfaces
Stream Data Collection Web Service
Static Data Collection Web Service
Temporal Abstraction Web Service
Relative Alignment Web Service
Exploratory Data Mining Web Service
Confirmatory Data Mining Web Service
Rule Management Web Service
Multi-Agent Data Mining
Processing Agent
Temporal Agent
Relative Agent$_n$
Relative Agent$_1$
Functional Agent$_n$
Functional Agent$_1$
Rules Generating Agent$_n$
Rules Generating Agent$_1$
Extended CRISP-DM Model
Data Underst anding
Data Preparation
Modelling
DM Ruleset Generation
Select Significant Ruleset
Formulate Null Hypothesis
Run Statistical Processes to test Hypothesis
Evaluation
Load accepted Rule-sets into RuleBase
STDM$^n_0$ Framework Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
Relative Alignment
Exploratory Data Mining across multiple streams for multiple patients
Confirmatory Data Mining with Null Hypothesis
Hypothesis/Rule generated and added to the Rulebase
Active Rules Ontology
Temporal Rules
Relative Rule
RuleBase Data
Static Data
Stream Data
Temporal Data
Relative Temporal Data
Study S$_1$
... Study S$_n$
Data Management

Figure 4.1 - The STDM$^n{}_0$ Framework

The framework also makes use of components from Foster and McGregor's (2005) multi-agent system (discussed in section 4.1.1), which has been extended to facilitate the tasks needed in the STDM$^n{}_0$ framework. In addition, the framework employs Heath's extended CRISP-DM data mining model (Heath, 2006, Heath and McGregor, 2010). The extended CRISP-DM model and the multi-agent system are integrated to allow for the new data mining model to complete the tasks of the STDM$^n{}_0$ framework.

1.1.1 Multi-Agent System

As discussed, the STDM$^n{}_0$ framework utilises the multi-agent system developed by Foster and McGregor (2005) which is an extension of the Analytical Processor that forms one of the components of the Solution Manager Service developed by McGregor (2005). The original framework consisted of an Agent Server which manages the communication between five agents: sub agent, processing agent, functional agent, rules generating agent and human agent. The framework also manages communication between the agents and the database access server (DBAS). The main agents utilized by the STDM$^n{}_0$ framework are the processing agent, temporal agent, relative agent, functional agent and rules generating agent (Bjering & McGregor, 2010).

1.1.2 Processing Agent

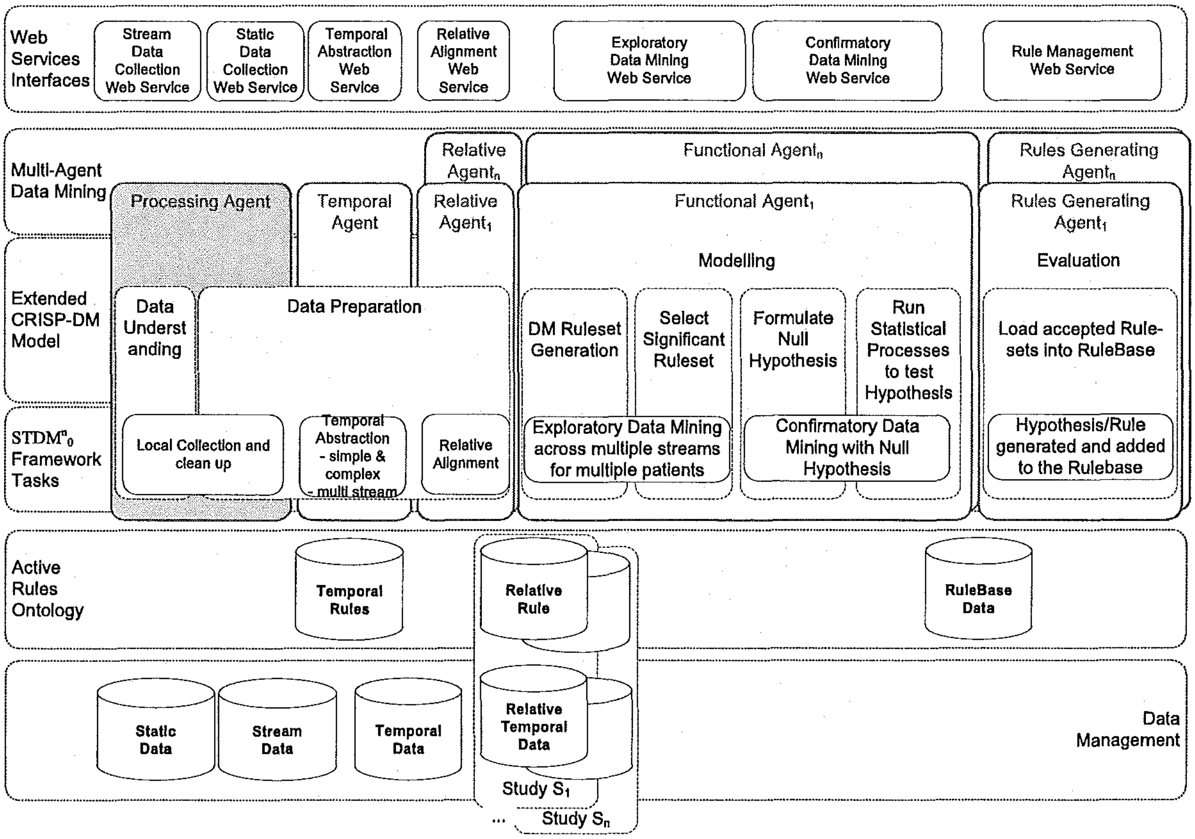
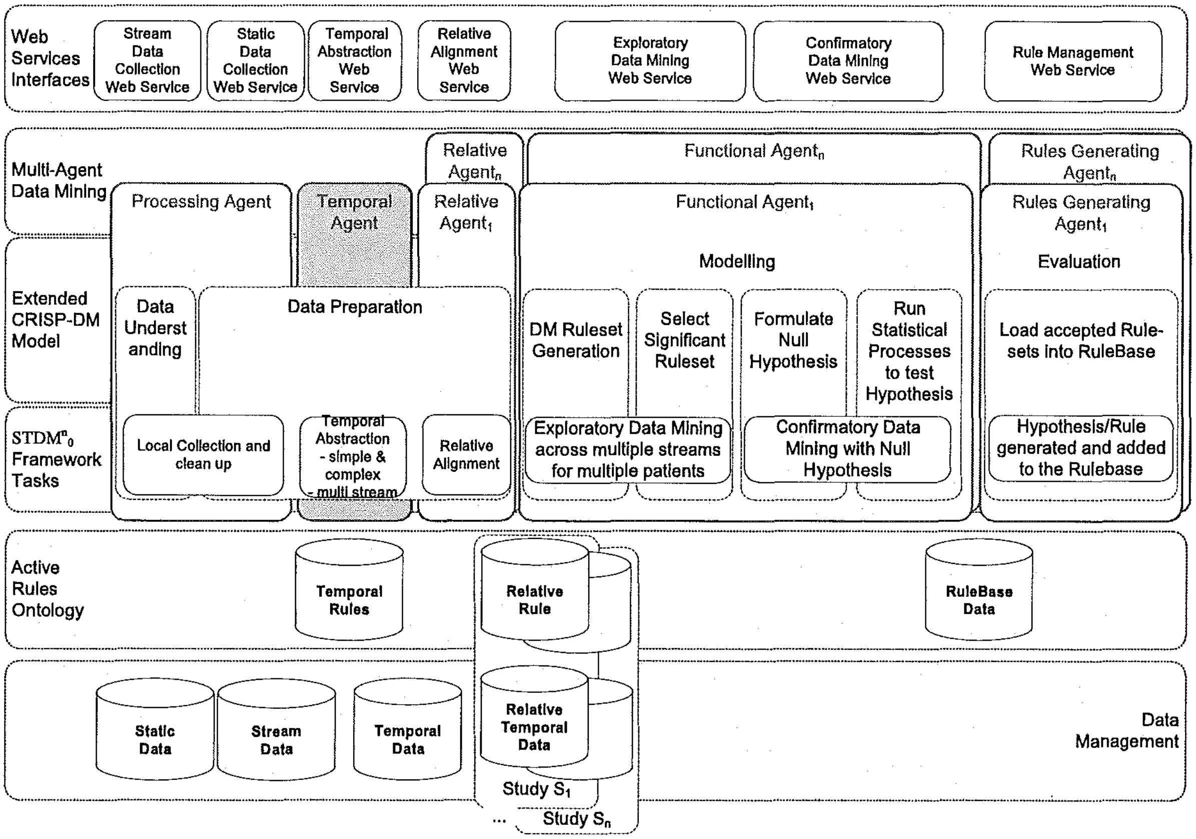

Web Services Interfaces
Stream Data Collection Web Service
Static Data Collection Web Service
Temporal Abstraction Web Service
Relative Alignment Web Service
Exploratory Data Mining Web Service
Confirmatory Data Mining Web Service
Rule Management Web Service
Multi-Agent Data Mining
Processing Agent
Temporal Agent
Relative Agent$_n$
Relative Agent$_1$
Functional Agent$_n$
Functional Agent$_1$
Modelling
Rules Generating Agent$_n$
Rules Generating Agent$_1$
Evaluation
Extended CRISP-DM Model
Data Understanding
Data Preparation
DM Ruleset Generation
Select Significant Ruleset
Formulate Null Hypothesis
Run Statistical Processes to test Hypothesis
Load accepted Rule-sets into RuleBase
STDM$^n{}_0$ Framework Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
Relative Alignment
Exploratory Data Mining across multiple streams for multiple patients
Confirmatory Data Mining with Null Hypothesis
Hypothesis/Rule generated and added to the Rulebase
Active Rules Ontology
Temporal Rules
Relative Rule
RuleBase Data
Static Data
Stream Data
Temporal Data
Relative Temporal Data
Study S$_1$
... Study S$_n$
Data Management

Figure 4.2 - The Processing Agent

The Processing Agent (Figure 4.2) is the first step into the STDM$^n{}_0$ framework; at this stage data is retrieved from external databases and processed in order to be stored in a

Physiological or Clinical Data Warehouse. This agent also supports the Data Understanding and Data Preparation phases of the extended CRISP-DM model and prepares the data to be used by the Temporal Agent for further processing.

1.1.3 Temporal Agent



Web Services Interfaces
Stream Data Collection Web Service
Static Data Collection Web Service
Temporal Abstraction Web Service
Relative Alignment Web Service
Exploratory Data Mining Web Service
Confirmatory Data Mining Web Service
Rule Management Web Service
Multi-Agent Data Mining
Relative Agent$_n$
Functional Agent$_n$
Rules Generating Agent$_n$
Processing Agent
Temporal Agent
Relative Agent$_1$
Functional Agent$_1$
Rules Generating Agent$_1$
Modelling
Evaluation
Extended CRISP-DM Model
Data Understanding
Data Preparation
DM Ruleset Generation
Select Significant Ruleset
Formulate Null Hypothesis
Run Statistical Processes to test Hypothesis
Load accepted Rule-sets into RuleBase
STDM$^{n}_{0}$ Framework Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
Relative Alignment
Exploratory Data Mining across multiple streams for multiple patients
Confirmatory Data Mining with Null Hypothesis
Hypothesis/Rule generated and added to the Rulebase
Active Rules Ontology
Temporal Rules
Relative Rule
RuleBase Data
Static Data
Stream Data
Temporal Data
Relative Temporal Data
Study S$_1$
...
Study S$_n$
Data Management

Figure 4.3 - The Temporal Agent

The Temporal Agent (Figure 4.3) utilizes physiological data being fed into the STDM$^{n}_{0}$ framework and helps create temporal abstractions based on the temporal rules defined in the temporal rules table. Temporal abstractions allow for the retention of the context of the data and act as a pre-processing method before data mining. They are also part of the data preparation phase of the extended CRISP-DM model. In order to better understand the movement and use of data at this stage, Bjering & McGregor (2010) outline the six main functions of the temporal agent below:

1. The first step is to retrieve the physiological data from the physiological data store for each parameter for each patient
2. Next, relevant abstraction rules are retrieved from the temporal rules table.
3. The rules are then applied to the physiological data, creating simple abstractions for individual data streams for individual patients
4. The created abstractions are then stored in the S***D***TDM$^{n}{}_{0}$ temporal data store
5. Complex abstractions are now created from the simple abstractions, based on any rules found in the temporal rules table.
6. Finally, any complex abstractions that are created are then stored in the S***D***TDM$^{n}{}_{0}$ temporal data store.

1.1.4 Relative Agent

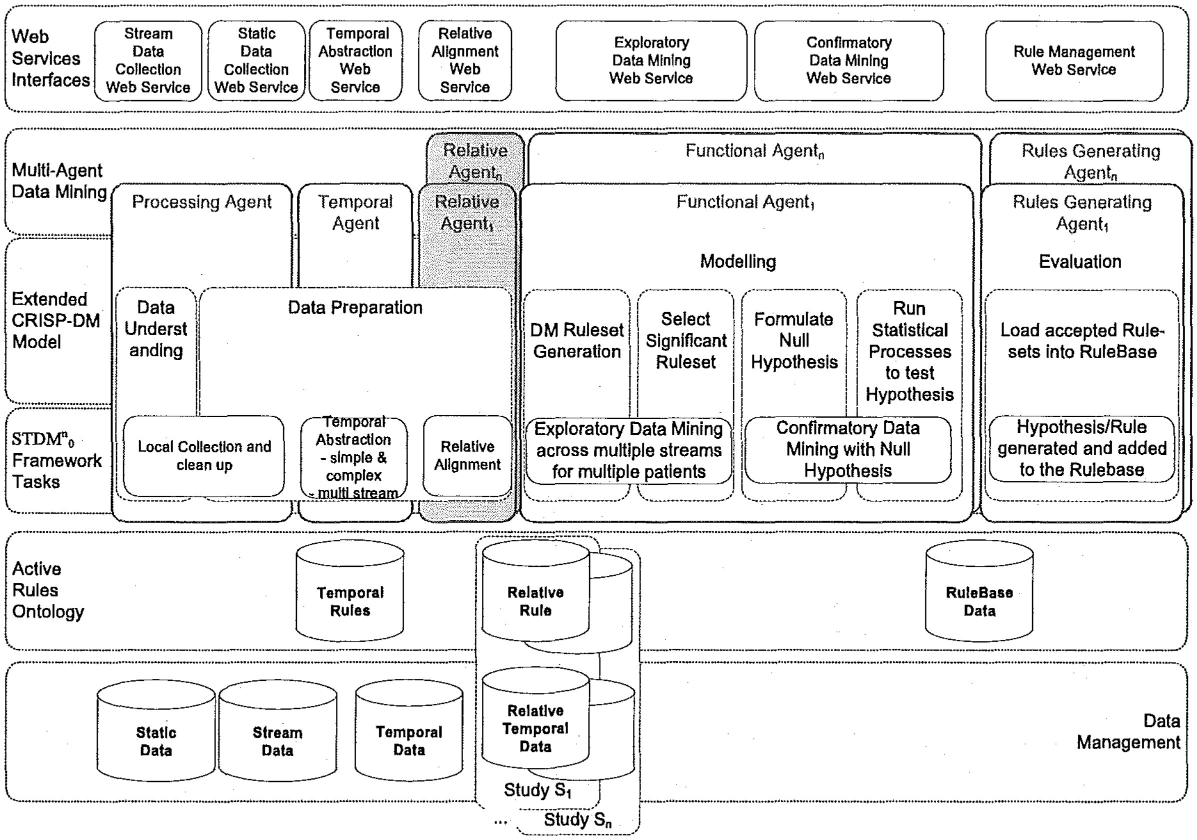

Web Services Interfaces
Stream Data Collection Web Service
Static Data Collection Web Service
Temporal Abstraction Web Service
Relative Alignment Web Service
Exploratory Data Mining Web Service
Confirmatory Data Mining Web Service
Rule Management Web Service
Multi-Agent Data Mining
Relative Agent$_n$
Functional Agent$_n$
Rules Generating Agent$_n$
Processing Agent
Temporal Agent
Relative Agent$_1$
Functional Agent$_1$
Rules Generating Agent$_1$
Modelling
Evaluation
Extended CRISP-DM Model
Data Underst anding
Data Preparation
DM Ruleset Generation
Select Significant Ruleset
Formulate Null Hypothesis
Run Statistical Processes to test Hypothesis
Load accepted Rule-sets into RuleBase
STDM$^n_0$ Framework Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
Relative Alignment
Exploratory Data Mining across multiple streams for multiple patients
Confirmatory Data Mining with Null Hypothesis
Hypothesis/Rule generated and added to the Rulebase
Active Rules Ontology
Temporal Rules
Relative Rule
RuleBase Data
Static Data
Stream Data
Temporal Data
Relative Temporal Data
Study S$_1$
... Study S$_n$
Data Management

Figure 4.4 - The Relative Agent

The Relative Agent (Figure 4.4) is used when dealing with clinical research studies. Once the Temporal Agent has created the abstractions from physiological data, it is common for this data to be used in various research studies. The Relative Agent uses the abstractions created by the Temporal Agent, together with clinical information from individual patients relative to the point of interest of the study, such as diagnosis of a particular clinical condition. The Relative Agent realigns the time of abstractions relative to a particular point in time that is of interest; this is an important step because the actual start and end times of the abstractions give no indication of what time this abstraction takes place in relation to the diagnosis of interest. This is done by calculating the start and finish times for each abstraction relative to a particular event. Finally, the abstractions that have been relatively aligned are stored in the relative temporal data store for further processing. It is also common for different research studies to use the same temporal abstractions which can lead to different realignment techniques to be applied to the same data. This is also the reason why every realigned temporal abstraction is stored in the relative temporal data table specific to the study that has utilized it.

1.1.5 Functional Agent

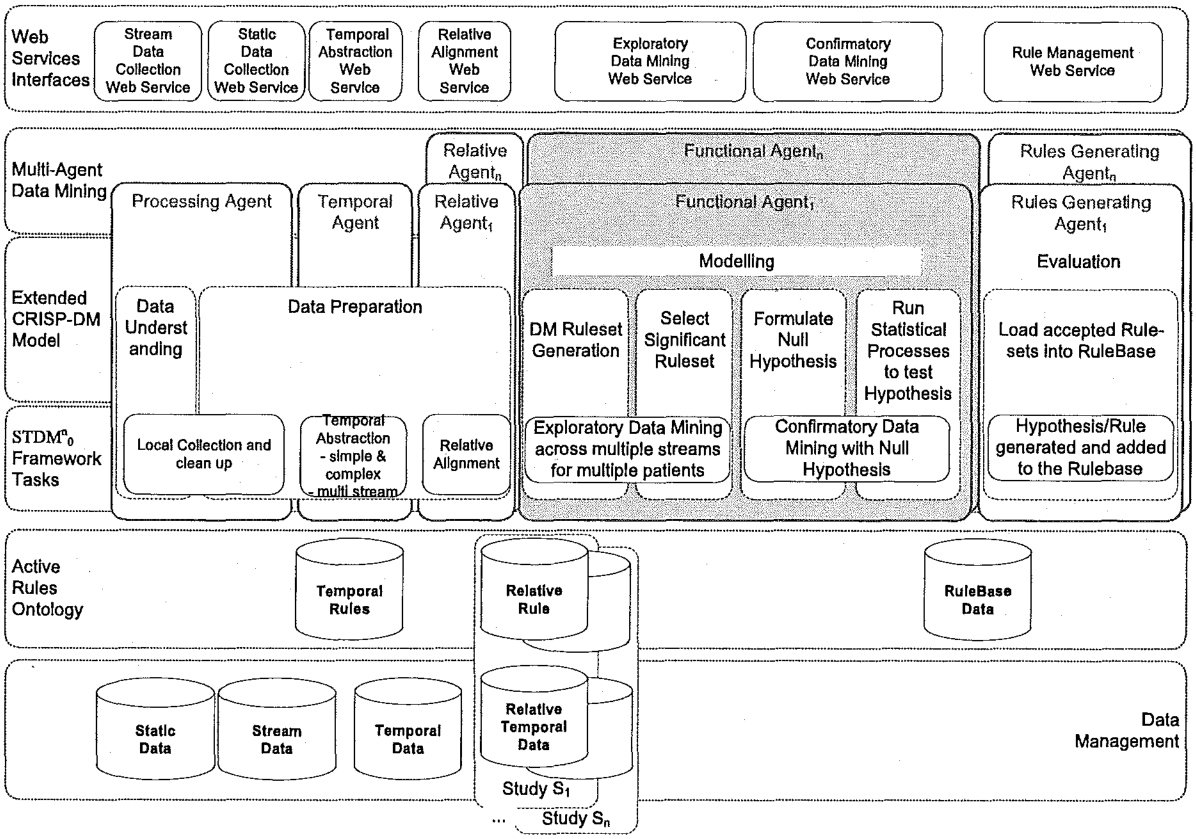

Web Services Interfaces
Stream Data Collection Web Service
Static Data Collection Web Service
Temporal Abstraction Web Service
Relative Alignment Web Service
Exploratory Data Mining Web Service
Confirmatory Data Mining Web Service
Rule Management Web Service
Multi-Agent Data Mining
Processing Agent
Temporal Agent
Relative Agent$_n$
Relative Agent$_1$
Functional Agent$_n$
Functional Agent$_1$
Rules Generating Agent$_n$
Rules Generating Agent$_1$
Modelling
Evaluation
Extended CRISP-DM Model
Data Understanding
Data Preparation
DM Ruleset Generation
Select Significant Ruleset
Formulate Null Hypothesis
Run Statistical Processes to test Hypothesis
Load accepted Rule-sets into RuleBase
STDM$^n_0$ Framework Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
Relative Alignment
Exploratory Data Mining across multiple streams for multiple patients
Confirmatory Data Mining with Null Hypothesis
Hypothesis/Rule generated and added to the Rulebase
Active Rules Ontology
Temporal Rules
Relative Rule
RuleBase Data
Static Data
Stream Data
Temporal Data
Relative Temporal Data
Study S$_1$
... Study S$_n$
Data Management

Figure 4.5 - The Functional Agent

The realigned temporal abstractions form the basis for exploratory and confirmatory data mining, processed by the Functional Agent (Figure 4.5). Exploratory data mining is used to analyse the realigned temporal abstractions across multiple data streams for multiple patients in order to detect new trends and patterns that might be present in the data prior to or after the event of interest. This also allows for the selection of the rules of significance based on the results of the exploratory data mining exercise. The next phase of confirmatory data mining begins with the formulation of the null hypothesis for any results that arouse interest and further investigation. The role of the confirmatory data mining process is to help prove or disprove the null hypothesis once it has been defined.

1.1.6 Rules Generating Agent

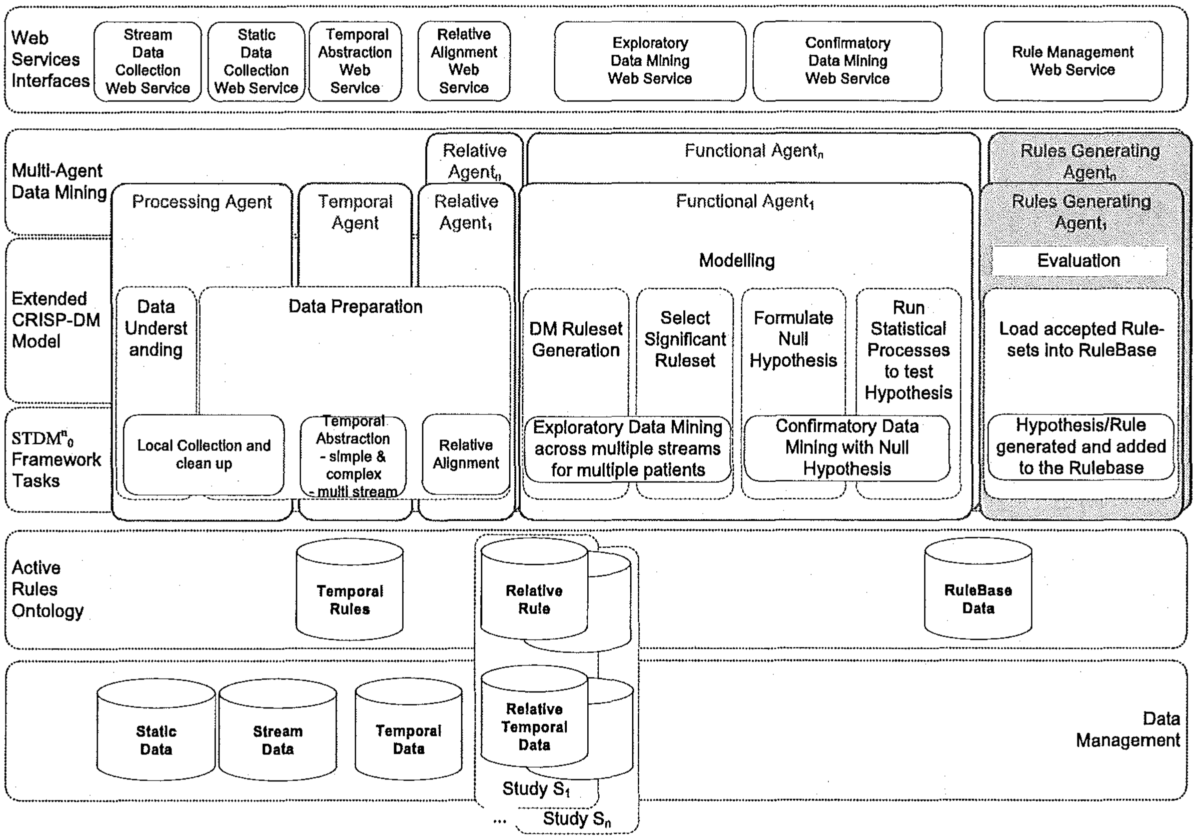

Web Services Interfaces
Stream Data Collection Web Service
Static Data Collection Web Service
Temporal Abstraction Web Service
Relative Alignment Web Service
Exploratory Data Mining Web Service
Confirmatory Data Mining Web Service
Rule Management Web Service
Multi-Agent Data Mining
Relative Agent$_n$
Functional Agent$_n$
Rules Generating Agent$_n$
Processing Agent
Temporal Agent
Relative Agent$_1$
Functional Agent$_1$
Rules Generating Agent$_1$
Modelling
Evaluation
Extended CRISP-DM Model
Data Understanding
Data Preparation
DM Ruleset Generation
Select Significant Ruleset
Formulate Null Hypothesis
Run Statistical Processes to test Hypothesis
Load accepted Rule-sets into RuleBase
STDM$^n_0$ Framework Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
Relative Alignment
Exploratory Data Mining across multiple streams for multiple patients
Confirmatory Data Mining with Null Hypothesis
Hypothesis/Rule generated and added to the Rulebase
Active Rules Ontology
Temporal Rules
Relative Rule
RuleBase Data
Static Data
Stream Data
Temporal Data
Relative Temporal Data
Study S$_1$
... Study S$_n$
Data Management

Figure 4.6 - The Rules Generating Agent

The Rules Generating Agent (Figure 4.6) utilizes findings made by the Functional Agent to allow for the creation of rules that can be defined in the rules database (Foster and McGregor 2005). The hypotheses created via the exploratory data mining phase are used by the Rules Generating Agent to create rules that can be stored and utilized by an event stream processor. This processor allows for the application of abstractions on real-time data streams which in turn can help establish these rules in a live analytical system to aid clinicians in real time analysis of data and provide alerts when necessary.

1.2 Implications of this research

One of the main themes that emerged from the review of existing literature in the area of data mining and distribution was a redundancy in the distribution approach. Commonly, the tasks that were distributed were used as temporary steps towards the data update process with the overall structure or framework still relying on a local storage and update for data and knowledge. We also have to consider the fact that a distributed database may contain a homogenous data set where the attributes describing the data are the same across each distributed database or a heterogeneous data set in which the attributes describing the data may differ.

Section 4.1 presented a detailed overview of the existing framework and outlined the processes involved at each of the Agents in the $STDM^n_0$ framework. However, the existing $STDM^n_0$ framework does not address the area of data distribution and lacks a structure which can support multicenter studies. The main limitations of the current framework include:

1. Notion of only one Temporal Rule table which is not suited for a multi centered approach.

2. Notion of only one Relative Rule table which is not suited for a multi centered approach.
3. Lack of a structure to accommodate multi centered studies, which may allow for the possibility of cross comparison of results between similar studies taking place at the same time.
4. Lack of clarity on how the Temporal Abstractions will be kept consistent in different locations/sites.
5. No discussion on how static and stream data can be handled in a distributed environment as this data set mostly contains patient identifying information which may not be easy to distribute due to improved privacy policies.

When we consider the possibility of a multidimensional application of the STDM$^{n}_{0}$ framework, it is clear that there is a need for a new approach towards the distribution of certain tasks, such as the Temporal Abstractions and temporal rules as well as the Relative and Functional Rules. There is a lack of clarity of how the distributed tasks will be performed and how they affect the systems at different sites.

This chapter has introduced the existing framework and supported the motivation of hypotheses 1, 2 and 3:

1. A multidimensional distributed data mining framework can be defined for time series data research for the discovery of trends and patterns prior to a given clinical event.
2. The framework will utilize elements of data fusion and agent-based analysis so that it will work with relational databases and large scale data mining applications.

3. A set of data mining tools can be applied for temporal abstraction, relative alignment and cluster analysis in a distributed manner to support multiple research studies.

5. Chapter 5 - Service Based Multi-Dimensional Distributed Temporal Data Mining (S*D*TDM$^n{}_0$)

The Service Based Multi-Dimensional Distributed Temporal Data Mining (S***D***TDM$^n{}_0$) provides the functionality determined as lacking in the STDM$^n{}_0$ framework (Figure 5.1), as discussed in Chapter 4. This chapter proposes a multidimensional distributed data mining framework that provides a structure to support multi center studies and manages the Temporal and Relative Rule tables in a distributed environment while maintaining consistency across the distributed sites.

The chapter addresses the following research hypotheses:

1. A multidimensional distributed data mining framework can be defined for time series data research for the discovery of trends and patterns prior to a given clinical event.
2. A set of data mining tools can be applied for temporal abstraction, relative alignment and cluster analysis in a distributed manner to support multiple research studies.
3. A set of data mining tools can be applied for temporal abstraction, relative alignment and cluster analysis in a distributed manner to support multiple research studies.

This chapter addresses the research hypotheses above by presenting a multidimensional distributed data mining framework that is suitable for use in clinical research, as shown in Figure 5.1. This framework addresses the limitations of the STDM$^n{}_0$ discussed in Chapter 4.

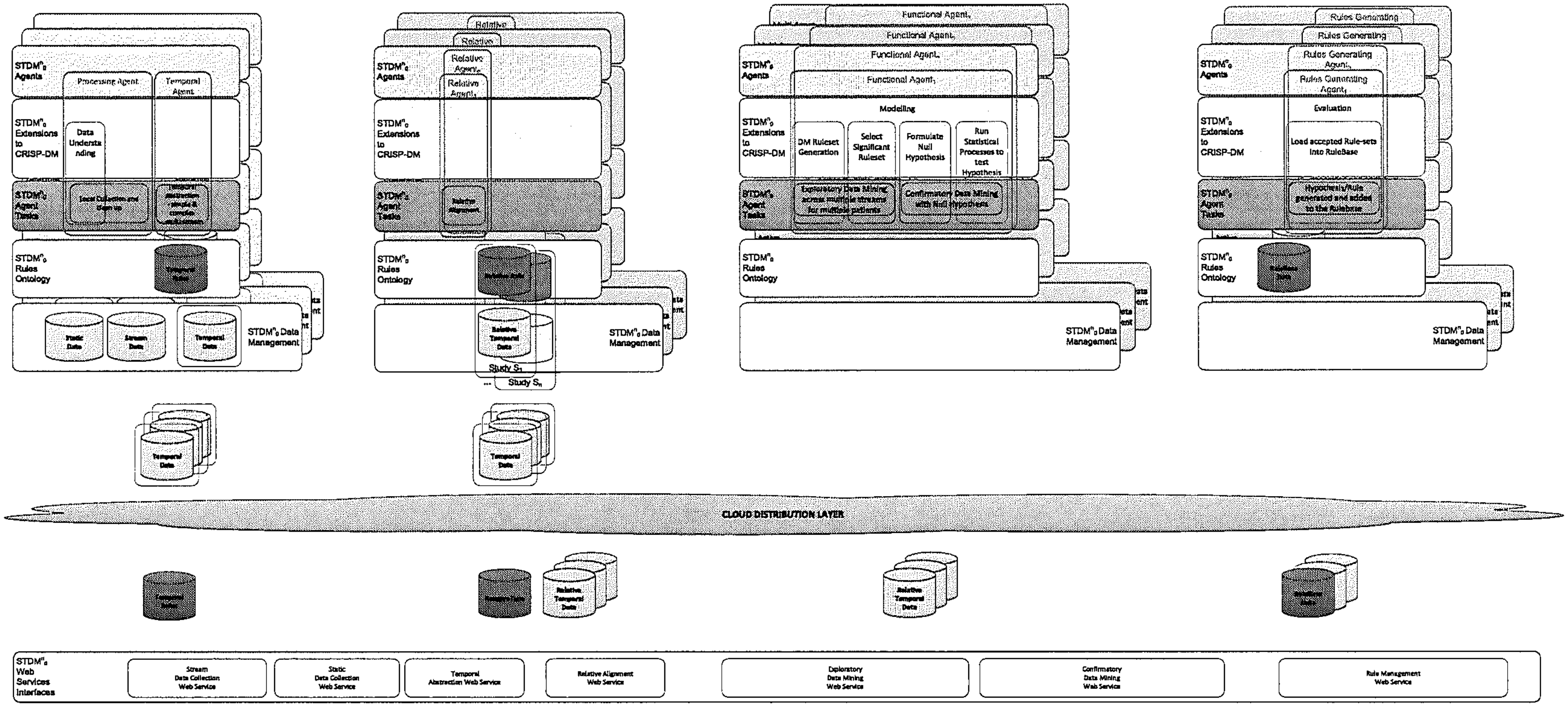

STDM$^{n}_{0}$ Agents
Processing Agent
Temporal Agent
STDM$^{n}_{0}$ Extensions to CRISP-DM
Data Understanding
STDM$^{n}_{0}$ Agent Tasks
Local Collection and clean up
Temporal abstraction - simple & complex multi-stream
STDM$^{n}_{0}$ Rules Ontology
Temporal Rules
Static Data
Stream Data
Temporal Data
STDM$^{n}_{0}$ Data Management
Relative
Relative
Relative Agent$_{n}$
Relative Agent$_{1}$
Relative Alignment
Relative Rules
Relative Temporal Data
Study S$_{1}$
... Study S$_{n}$
Temporal Data
Temporal Data
Functional Agent$_{n}$
Functional Agent$_{j}$
Functional Agent$_{i}$
Functional Agent$_{1}$
Modelling
DM Ruleset Generation
Select Significant Ruleset
Formulate Null Hypothesis
Run Statistical Processes to test Hypothesis
Exploratory Data Mining across multiple streams for multiple patients
Confirmatory Data Mining with Null Hypothesis
Rules Generating
Rules Generating
Rules Generating Agent$_{n}$
Rules Generating Agent$_{1}$
Evaluation
Load accepted Rule-sets into RuleBase
Hypothesis/Rule generated and added to the Rulebase
RuleBase
CLOUD DISTRIBUTION LAYER
Temporal Rules
Relative Rules
Relative Temporal Data
Relative Temporal Data
STDM$^{n}_{0}$ Web Services Interfaces
Stream Data Collection Web Service
Static Data Collection Web Service
Temporal Abstraction Web Service
Relative Alignment Web Service
Exploratory Data Mining Web Service
Confirmatory Data Mining Web Service
Rule Management Web Service

Figure 5.1 - The SDTDM$^{n}_{0}$ Framework

5.1 The Distributed Temporal Agent

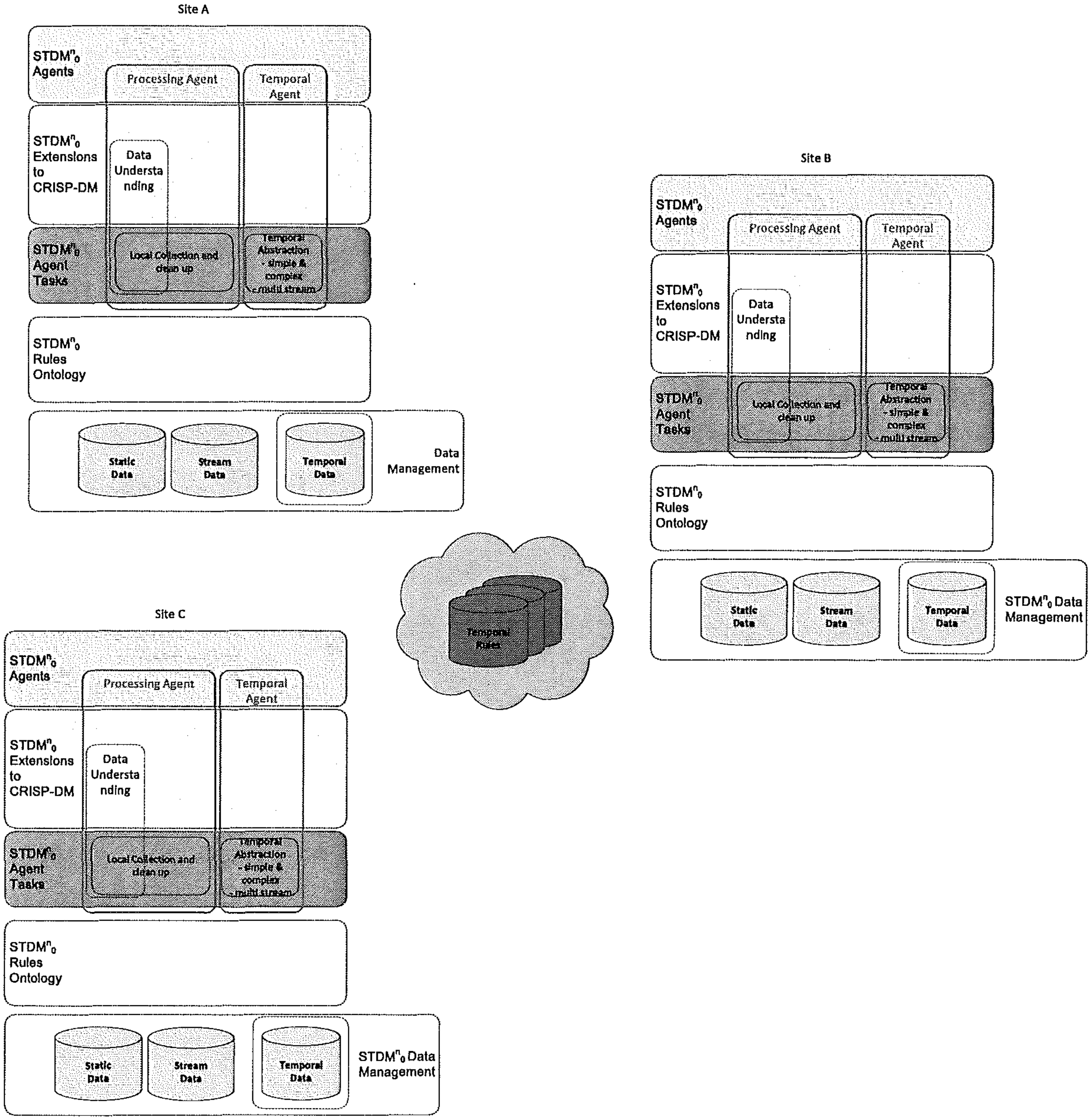

Site A
STDM$^n_0$ Agents
Processing Agent
Temporal Agent
STDM$^n_0$ Extensions to CRISP-DM
Data Understanding
STDM$^n_0$ Agent Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
STDM$^n_0$ Rules Ontology
Static Data
Stream Data
Temporal Data
Data Management
Site B
STDM$^n_0$ Agents
Processing Agent
Temporal Agent
STDM$^n_0$ Extensions to CRISP-DM
Data Understanding
STDM$^n_0$ Agent Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
STDM$^n_0$ Rules Ontology
Static Data
Stream Data
Temporal Data
STDM$^n_0$ Data Management
Temporal Rules
Site C
STDM$^n_0$ Agents
Processing Agent
Temporal Agent
STDM$^n_0$ Extensions to CRISP-DM
Data Understanding
STDM$^n_0$ Agent Tasks
Local Collection and clean up
Temporal Abstraction - simple & complex - multi stream
STDM$^n_0$ Rules Ontology
Static Data
Stream Data
Temporal Data
STDM$^n_0$ Data Management

Figure 5.2 - The Distributed Temporal Agent

As discussed in Chapter 4, the Temporal Agent manages physiological data being used by the STDM$^n_0$ framework and helps create temporal abstractions based on the temporal rules.

The main elements in this phase are the creation of the simple abstractions for individual data streams for individual patients which are stored in the $STDM^n_0$ temporal data store and the creation of complex abstractions based on any rules found in the temporal rules table which are also stored in the $STDM^n_0$ temporal data store.

Based on the existing architecture, the creation and storage of Temporal Abstractions and Temporal Rules are local to each site and have no mechanism for distribution. In a multidimensional environment, the physiological data that is being retrieved would come from multiple sites which may not be the same in terms of data structure or even data frequency. It would also not be very efficient to have multiple local stores of temporal data and temporal rules for each site. However, due to current health care policies and improved patient privacy concerns, it is required that the static and stream data as well as the Temporal Abstractions exist locally at each site. The Temporal Rules, however, do not contain patient identifying information and thus can be decentralized (Figure 5.2). Several advantages arise from de-centralizing data:

1. Allows for the Temporal Abstractions and Rules to be kept consistent across different sites.
2. Allows for better control over the security of the data as there is only one location to manage.
3. Allows for better accessibility to the data through a controlled and secure environment.
4. A decentralized environment is very modular with respect to resource management.

The task of decentralizing the Temporal Rules starts by moving elements of the framework into the central data server that will act as a cloud distribution layer across all participating sites. The following four steps describe the distributed approach in detail:

1. The physiological data is retrieved from the physiological data store for each parameter for each patient.
2. A link is made with the cloud distribution layer in order to retrieve the relevant abstraction rules from the temporal rules table which are then applied to the physiological data.
3. The simple abstractions that are created for individual data streams for individual patients are then stored locally at each site. They are also tagged with a SITE_ID for ease of identification of their source site for comparison studies.
4. Complex abstractions are created from the simple abstractions using the temporal rules table. Once completed the newly created complex abstractions are also stored locally in the same TA tables and tagged for easy identification.

5.2 The Distributed Relative Agent

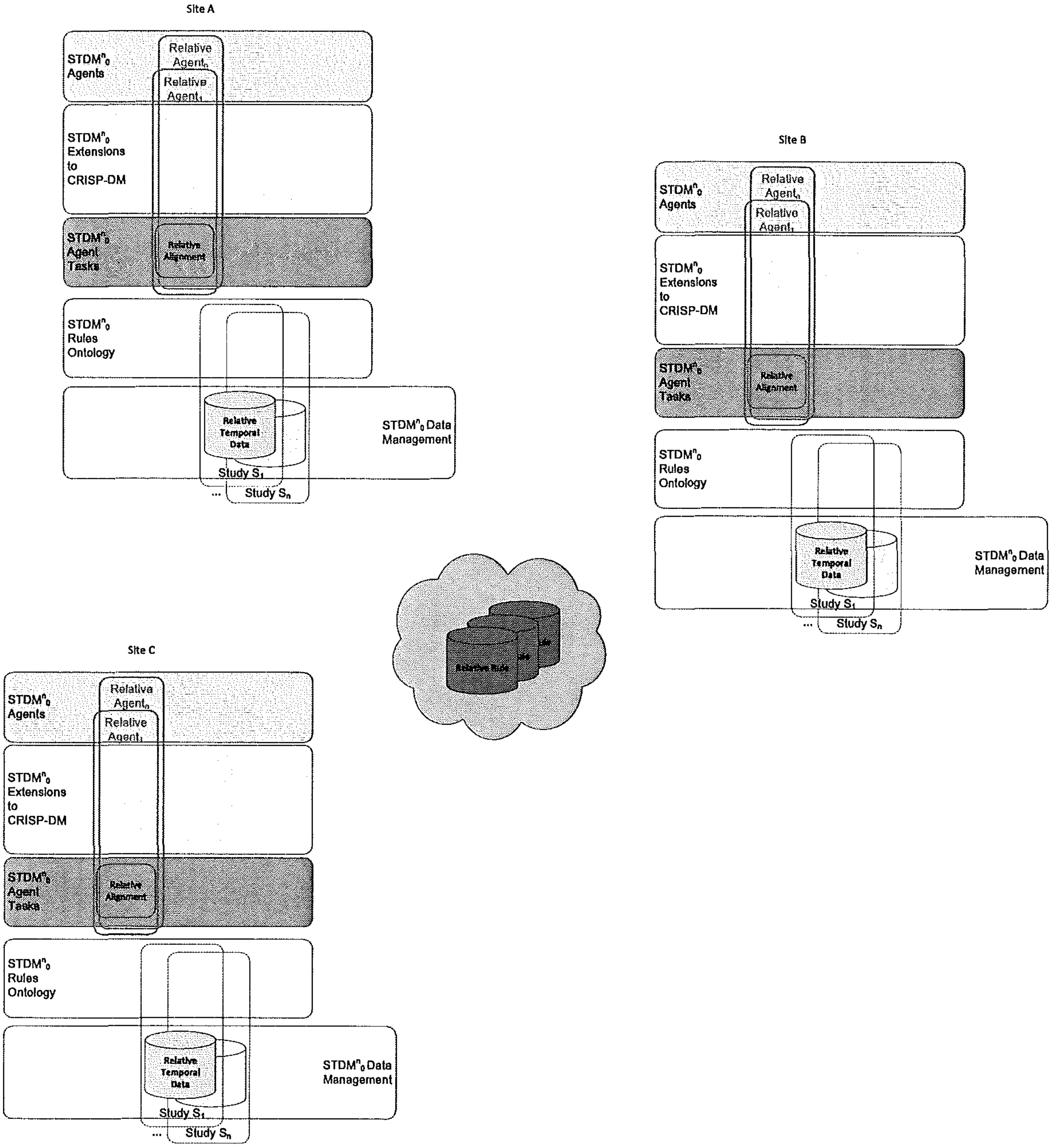


Figure 5.3 - The Distributed Relative Agent

The Relative Agent plays an important role in clinical research studies and can greatly benefit from a distributed framework. The Relative Agent needs access to the abstractions created by the Temporal Agent, as well as the clinical information of the individual patient relative to the time of the study of interest. In order to enable this functionality in a distributed structure, the distributed framework makes use of the Relative Alignment Web Service which acts as the gatekeeper for data access. It is important to note that different research studies might use the same temporal abstractions. For this reason, the central data server will contain a relative temporal data table specific to each study. Abstractions that have been relatively aligned can be stored in the relative temporal data store and tagged for easy identification as well.

By decentralizing the Relative Rule data (Figure 5.3) from the original framework we can enable multicenter studies to take place simultaneously and also allow for the possibility of cross comparison of results between similar studies taking place at the same time.

5.3 The Distributed Functional & Rules Generating Agents

The functional agent performs data mining tasks used to enable detection of interesting trends and patterns for a particular study. Exploratory data mining is used to analyse the realigned temporal abstractions across multiple data streams for multiple patients in order to detect new trends and patterns that might be present in the data prior to or after the event of interest. Once possible trends and patterns have been discovered, they need to be evaluated by the clinician to enable the creation of a hypothesis. This also allows for the selection of the rules of significance based on the results of the exploratory data mining exercise. The steps involved in the distribution of these agents are as follows (represented diagrammatically in Figure 5.4):

1. Exploratory Mining used to analyse the realigned temporal abstractions, from the Relative Agent, across multiple data streams for multiple patients in order to detect new trends and patterns that might be present in the data.

2. Rules Generating Agent uses the exploratory functional rules in the creation of a new Rule Base Data table which is then stored centrally.
3. Event Stream Processor connects with Rule Base Data table for the application of abstractions on real-time data streams.

Details of the local stores seen in Figure 5.4 will be discussed in greater detail in Chapter 6.

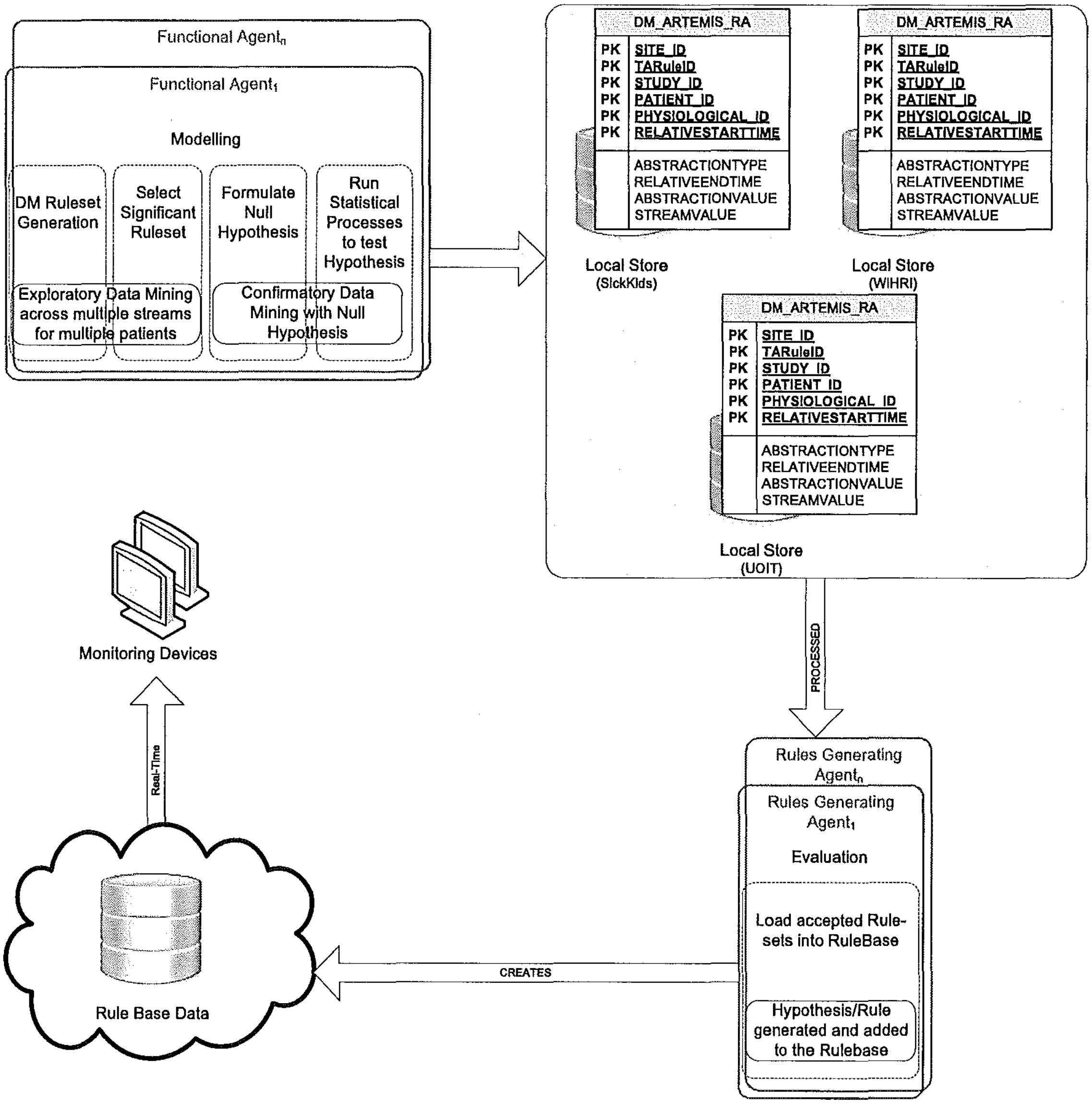


Figure 5.4 - The Distributed Functional & Rules Generating Agents

5.4 Data Structure & Storage

The overall data storage schema for the STDM$^n{}_0$ can be seen in Figure 5.5 (McGregor C. P., 2010). This section will elaborate on the components being distributed and the changes that will take place.

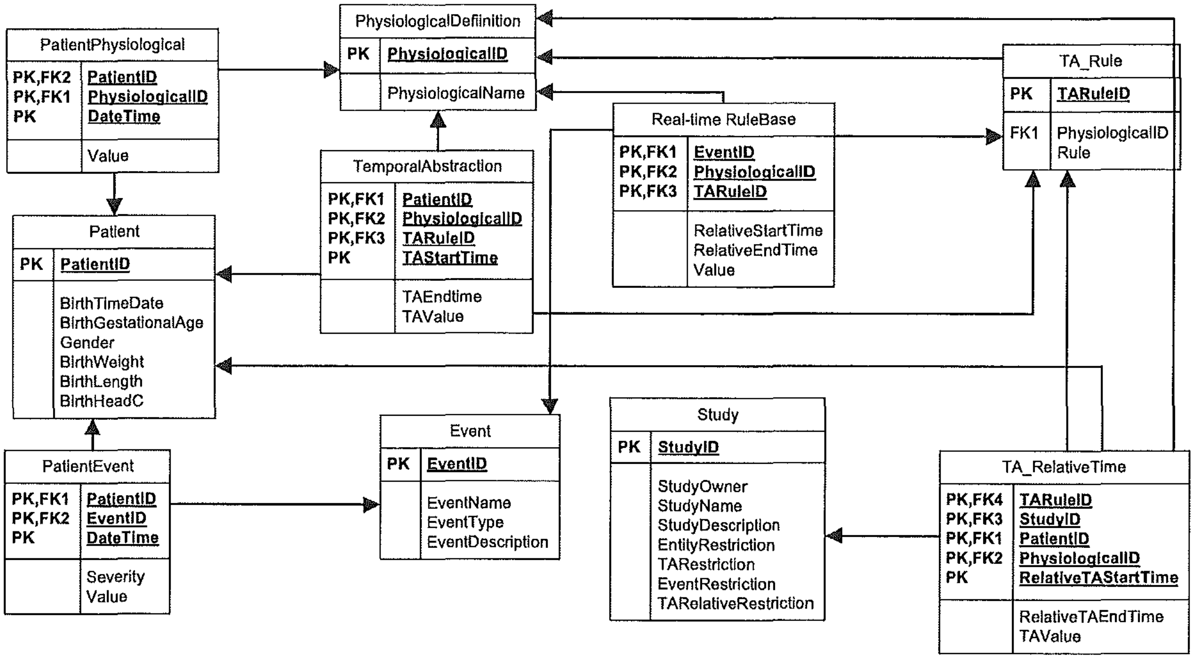


Figure 5.5 - The STDM$^n{}_0$ Data Storage Schema

5.4.1 Temporal Rules

The TA_Rule table (Figure 5.6) contains rules for how to abstract particular physiological parameters. Each physiological parameter can be linked to multiple rules and incidentally create more than one abstraction. The TA_Rule table is also capable of containing the entire SQL abstraction query that needs to be run to abstract particular physiological parameters.

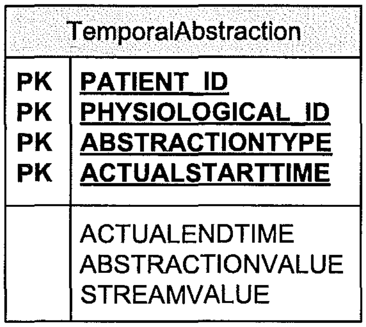
TA_Rule	
PK	**TARuleID**
FK1	PhysiologicalID Rule

Figure 5.6 - The TA_Rule Table

There are three attributes in the TA_Rule table i.e. the TARuleID which contains the unique ID of a particular rule, the PhysiologicalID which links the TA_Rule table to the PhysiologicalDefinition table, and is used to identify which type of parameter the particular rule applies to and finally the Rule attribute which contains the details of the particular rule including the SQL query needed to run the rule. The TA_Rule table has a many to one relationship to the PhysiologicalDefinition table, which indicates that a particular PhysiologicalDefinition can have more than one TA rule applied to it.

5.4.2 Temporal Abstraction Data

The Temporal Abstraction table (Figure 5.7) contains TAs created from the patient's physiological parameters (McGregor C. P., 2010). The temporal abstractions stored in this table are created by applying the rules contained in the TA_Rule table to the relevant physiological parameter of a patient. The PatientID attribute is used to link a particular abstraction to a particular patient, the PhysiologicalID attribute is used to relate the abstraction to a particular physiological definition, ABSTRACTIONTYPE indicates the type of abstraction i.e. a trend or a level shift, ABSTRACTIONVALUE contains the results of the abstraction. These values may indicate an increase, decrease or a range of values from high to normal (McGregor C. P., 2010). Finally, the ACTUALSTARTTIME and ACTUALENDTIME attributes indicate the time that the abstraction became true and when the particular abstraction no longer holds true. This table is linked to the Patient table in a many to one relationship, thus implying that a particular patient can have many abstractions stored in the table. The Temporal Abstraction table also maintains a many to one relationship with the PhysiologicalDefinition table meaning that a physiological parameter can have several abstractions performed on it.

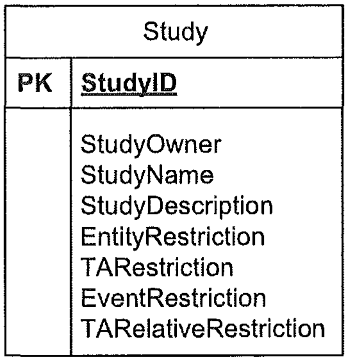


TemporalAbstraction	
PK	**PATIENT_ID**
PK	**PHYSIOLOGICAL_ID**
PK	**ABSTRACTIONTYPE**
PK	**ACTUALSTARTTIME**
	ACTUALENDTIME ABSTRACTIONVALUE STREAMVALUE

Figure 5.7 - The Temporal Abstraction Table

5.4.3 Relative Rule

The Relative Rule or Study table (Figure 5.8) specifies a particular alignment of abstractions for a particular study and contains the information about any relative rules that may need to be applied to the abstractions stored in the Temporal Abstraction table (McGregor C. P., 2010). The StudyID attribute is a unique identifier for each study. The StudyOwner, StudyName and StudyDescription attributes contain details on the study and to whom they belong. The EntityRestriction, TARestriction, EventRestriction and the TARelativeRestriction attributes contain where clauses providing higher levels of constraints to the Study table (McGregor C. P., 2010).

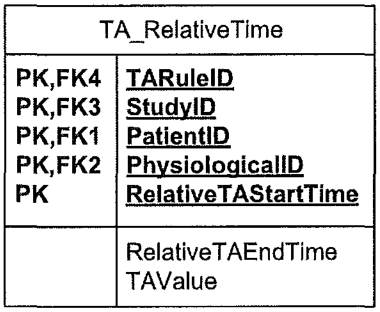


Study	
PK	**StudyID**
	StudyOwner StudyName StudyDescription EntityRestriction TARestriction EventRestriction TARelativeRestriction

Figure 5.8 - The Relative Rule Table

5.4.4 Relative Temporal Abstractions

The Relative Temporal Abstraction table (Figure 5.9) holds the abstractions that have been realigned relative to a point of interest to the researcher who owns the study (McGregor C. P., 2010). The attributes of this table are similar to the Temporal Abstraction table (Figure 5.7) except that this table contains RelativeTAStartTime and RelativeTAEndTime values which are times relative to the period in time that is interesting to the researchers/owners of the study. A unique StudyID attribute is also included in this table to allow abstractions to be linked with the Study Table (Figure 10) with which it shares a many to one relationship meaning that there can be many entries in the TA_RelativeTime table that belong to a particular study (McGregor C. P., 2010).

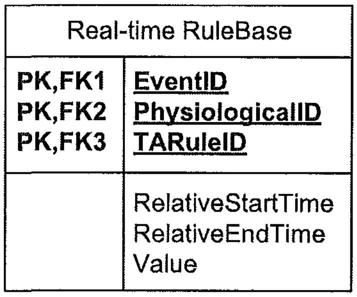


TA_RelativeTime	
PK,FK4 PK,FK3 PK,FK1 PK,FK2 PK	**TARuleID** **StudyID** **PatientID** **PhysiologicalID** **RelativeTAStartTime**
	RelativeTAEndTime TAValue

Figure 5.9 - The Relative Temporal Data Table

5.4.5 Rule Base Data

The Rules created from hypothesis as a result of the last step can then be stored in the RuleBase table (Figure 5.10). The attributes of this table include a unique EventID, PhysiolocialID and TARuleID from the tables discussed earlier as well as the RelativeStartTime and RelativeEndTime attributes (McGregor C. P., 2010). A Value attribute is also contained in this table indicating the threshold values that are of interest to researchers and can be deployed in a real time environment, for example, a lapse in the breathing of a neonate for greater than 15 seconds and a fall in peripheral oxygen saturation less than 85% for greater than 20 seconds combined with a heart rate of less than 100 BPM may be an indicator of an apneic event (Catley et. al. 2010).



Real-time RuleBase	
PK,FK1 PK,FK2 PK,FK3	EventID PhysiologicalID TARuleID
	RelativeStartTime RelativeEndTime Value

Figure 5.10 - The Rule Base Table

5.6 Design Changes to Support Distributed Functionality

In order to perform temporal abstractions on data, the data must first be processed from its raw format. The role of the processing agent is to initiate collection of stored physiological and clinical data from external data stores supporting online analysis. Once the data has passed from the external collection phase, the Processing Agent converts the data to the required format, if necessary, and then the data is structured and stored in the clinical data and physiological data tables. Once completed, the Temporal Agent begins to process data in order to create the temporal abstractions using rules defined in the temporal rules table.

5.6.1 Distribution of Temporal Rules

Chapter 4 presented the limitations of the $STDM^n_0$ framework, one of which was the fact that it is structured to support only one Temporal Rule table. There was also a lack of clarity on how the Temporal Abstractions will be kept consistent across the multidimensional distributed locations. Figure 5.11 presents a solution to this problem and highlights the management of multiple Temporal Rule tables while keeping the Temporal Abstractions consistent across the distributed sites. As discussed in Section 5.1, due to current health care policies and improved patient privacy concerns, it is required that certain types of data exist locally at each site. Thus the Temporal Abstractions have to be stored locally at each distributed site, but the same is not true for the Temporal Rules. As they contain no patient identifying information, they can be de-centralized to allow for consistency, better control over the security and better accessibility. Figure 5.11 presents a structure which supports the distribution of Temporal Rules and at the same time provides a solution for the Distribution of Temporal Abstractions where they are allowed to be distributed.

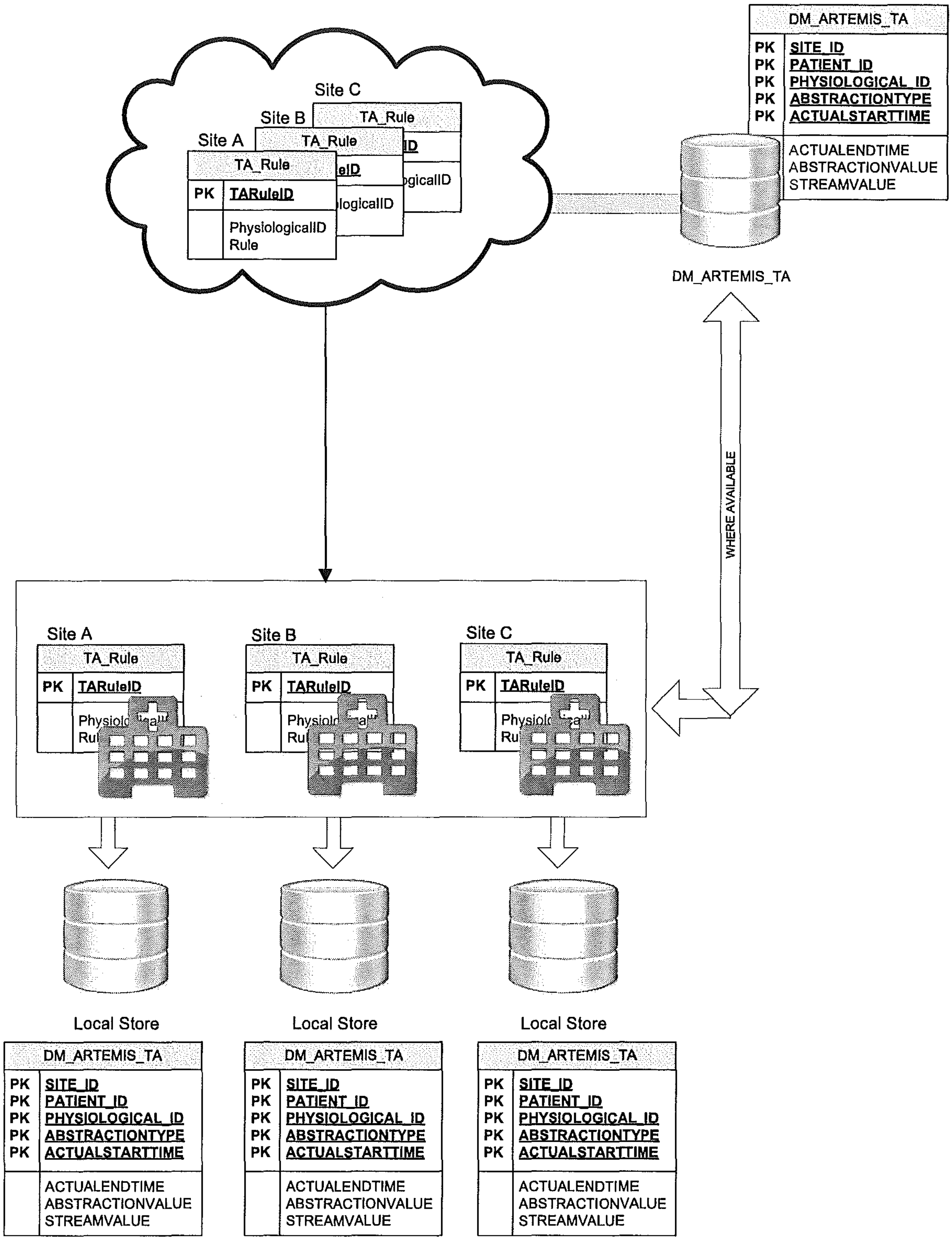


Figure 5.11 - Distribution of Temporal Rules

Policies regarding the handling of data and its privacy will always differ across the multidimensional distributed sites; hence the need to support the data in a distributed setting. Following are the steps involved in the distribution of Temporal Rules, as shown in Figure 5.11:

1. The Temporal Rules have been distributed so that they are stored centrally. When TA's need to be run, the associated rules are deployed simultaneously for each participating site (Site A, Site B, Site C ... Site N). The TA rules deployed for each site also contain the SQL query that needs to be run to perform the abstraction at each site as this is supported by the TA_Rule table.
2. Once deployed, the Temporal Abstractions are run locally at each site based on the parameters provided by the Temporal Rules.
3. A unique identifier is attached to the resulting output. This identifier is called the SITE_ID tag which gives each location a unique ID and also allows for comparison of results across sites when needed.
4. Finally, the results of the Temporal Abstractions are stored locally at each site in data tables (DM_ARTEMIS_TA). Where available, these results will also be populated back at the central data store (DM_ARTEMIS_TA).

5.6.2 Distribution of Relative Rules

The Relative Rule table, which specifies a particular alignment of abstractions for a particular study, holds the information about any relative rules that may need to be applied to the abstractions stored in the Temporal Abstraction table. In Chapter 4, it was highlighted that the existing $STDM^n_0$ framework supports the notion of only one Relative Rule table which is not suited for multidimensional distributed studies.

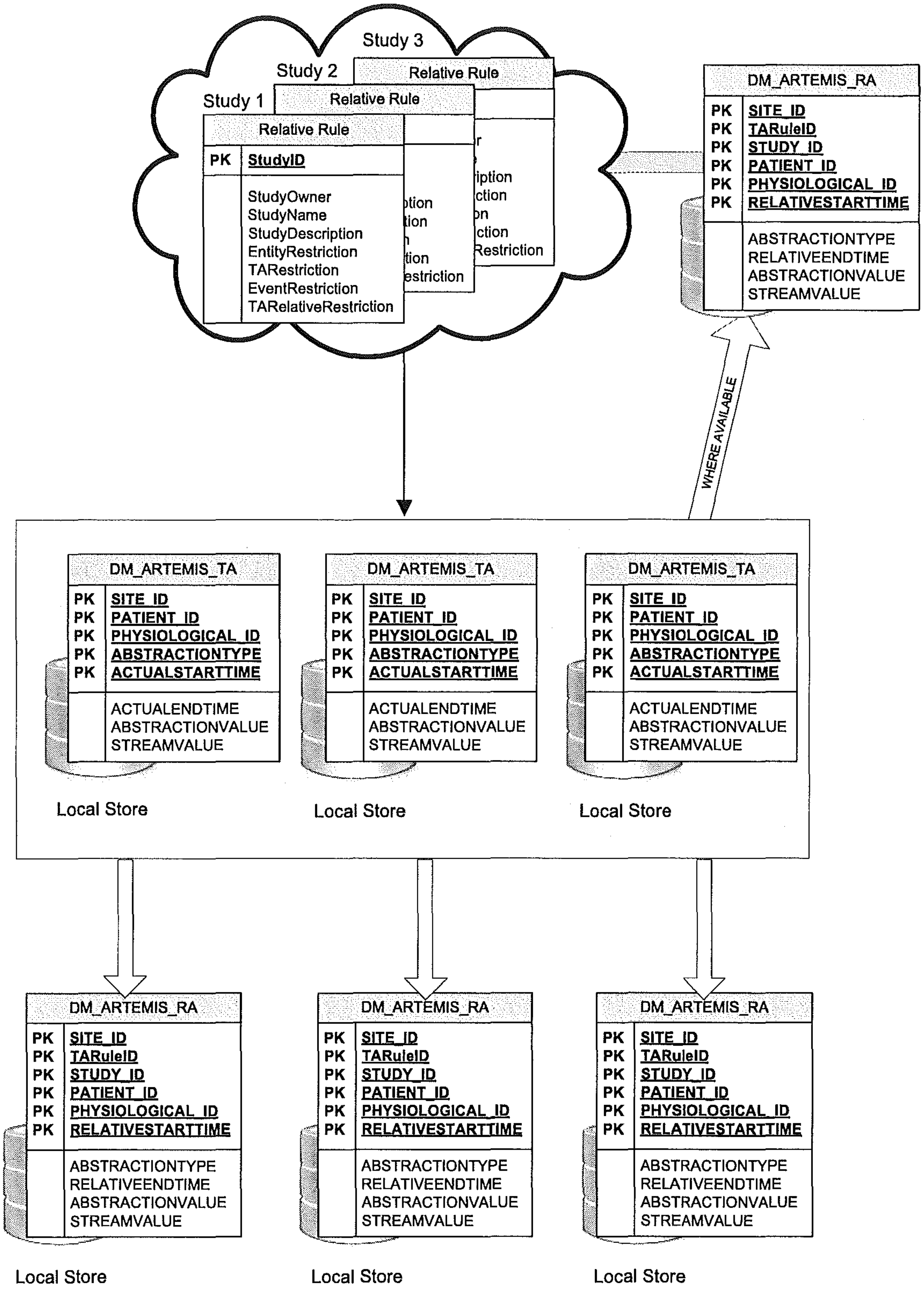


Figure 5.12 - Distribution of Relative Rules

Figure 5.12 outlines the structure supporting the distribution of Relative Rules. The following three step approach is taken to enable the distribution of Relative Rules:

1. Relative rules for each study are deployed from the central data store. A separate study table exists for each participating facility and is assigned a unique StudyID.
2. Once deployed, the Temporal Abstractions table created at each site is accessed locally in order to perform the Relative Alignments needed for the particular study.
3. The re-aligned Temporal Abstractions are then created and stored locally in the Relative Temporal Data tables specific to the study and the site which is identified by the unique SITE_ID (DM_ARTEMIS_RA). Where supported, the Relative Temporal Data will also be populated back at the central data store (DM_ARTEMIS_RA) with the addition of a SITE_ID tag that allows for separation and comparison between sites.

5.6.3 Distribution of Rule Base Data

The Rules Generating Agent utilizes findings made by the Functional Agent to allow for the creation of rules that can be defined in the rules database. Hypotheses created via the exploratory data mining phase are used by the rules generating agent to create rules that can be stored and utilized by an event stream processor in the application of abstractions on real-time data streams. The distributed Rule Base data exists centrally and is invoked every time a rule needs to be applied for real-time monitoring. In this case (Figure 5.4), the Functional Agent invokes the Relative Temporal Abstractions stored locally at each site (DM_ARTEMIS_RA). The Rules Generating Agent then uses results produced by the Functional Agent to create Rule Base Data in the central data store. These rules can then by deployed for active real-time monitoring of patient data.

5.7 Conclusion

This chapter has presented the design of the S***D***TDM$^{n}_{0}$ framework, which has been extended by this research to support a multidimensional distributed data mining environment. This allows for the management of Temporal and Relative Rule tables in a distributed environment to support multicenter studies as well as the distribution of Rule Base data which can be applied for real-time monitoring across sites. The chapter has addressed research hypotheses one, two and three by demonstrating and defining the S***D***TDM$^{n}_{0}$ framework, a multidimensional distributed data mining framework that is suitable for use in clinical research.

6. Chapter 6 - Demonstration within the NICU Context

This chapter presents a demonstration of the architectural framework detailed in Chapter 5 within the context of its deployment to support clinical research in neonatal intensive care. Evidence for research hypotheses three and four is presented with further validation to support hypotheses one and two. These hypotheses were:

1. A multidimensional distributed data mining framework can be defined for time series data research for the discovery of trends and patterns prior to a given clinical event.
2. The framework will utilize elements of data fusion and agent-based analysis so that it will work with relational databases and large scale data mining applications.
3. A set of data mining tools can be applied for temporal abstraction, relative alignment and cluster analysis in a distributed manner to support multiple research studies.
4. The framework can be applied in a broad neonatal context addressing issues of data privacy and confidentiality and being deployable as part of multicenter studies while maintaining data integrity at each participating site.

Through an active collaboration between The Hospital of Sick Children, Toronto, led by Dr. Andrew James, The Women and Infants Hospital (WIHRI), Providence, Rhode Island, led by Dr. James Padbury and the Health Informatics Research team, University if Ontario Institute of Technology (UOIT), Oshawa, led by Dr. Carolyn McGregor, we are utilising current clinical research activities within the NICU to demonstrate the architecture proposed in Chapter 5 and provide analytical support for the clinical research activities. The research being conducted at UOIT is part of the clinical research studies that have been ethically approved at both sites as part of the Artemis project. Artemis is a platform for real-time enactment of clinical knowledge as it relates to multi-dimensional data analysis and clinical research. The Artemis framework (as seen in Figure 6.1) is a platform for real-time analysis of clinical knowledge as it relates to multi-dimensional data analysis and clinical research.

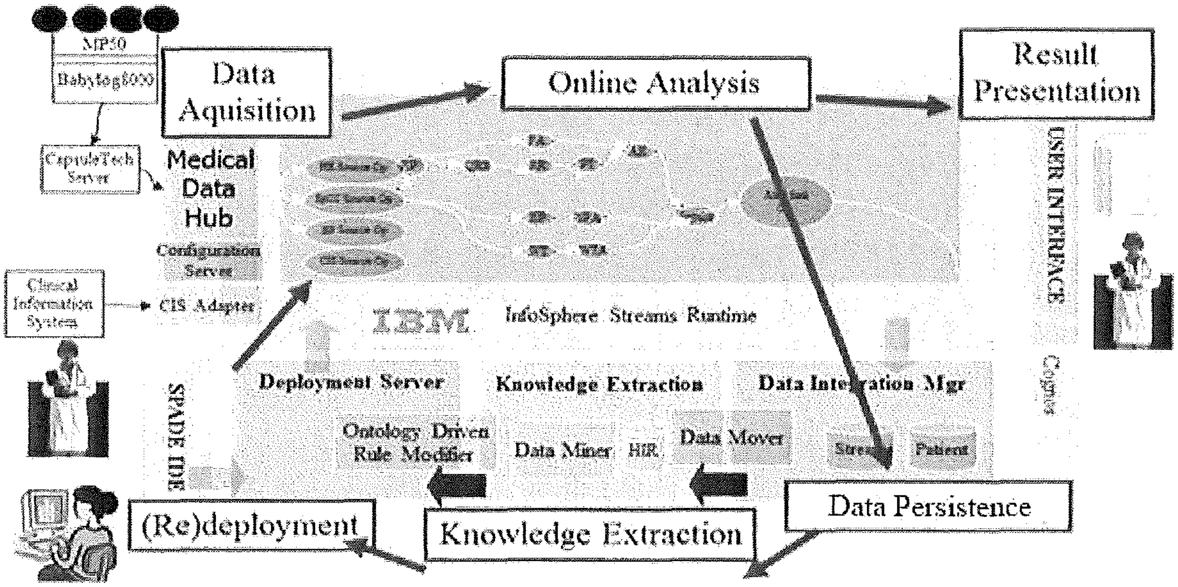


Figure 6.1 - The Artemis Framework (UOIT Health Informatics Research).

As discussed in Chapter 3, there is mounting evidence suggesting changes in physiological stream behaviours prior to the diagnosis of certain conditions. The Health Informatics Research group at UOIT is focusing on research into earlier detection of late onset neonatal sepsis and episodes of apnoea using physiological stream data being collected from three distributed sites. A number of parameters are being analysed in order to support this research such as: 1) abstractions for heart rate decelerations in an hourly time window; 2) fall in peripheral oxygen saturation less than 85% for greater than 20 seconds; 3) a lapse in breathing of a neonate of 35 weeks gestation for greater than 15 seconds; and 4) a low heart rate and respiratory rate variability in an hourly segment.

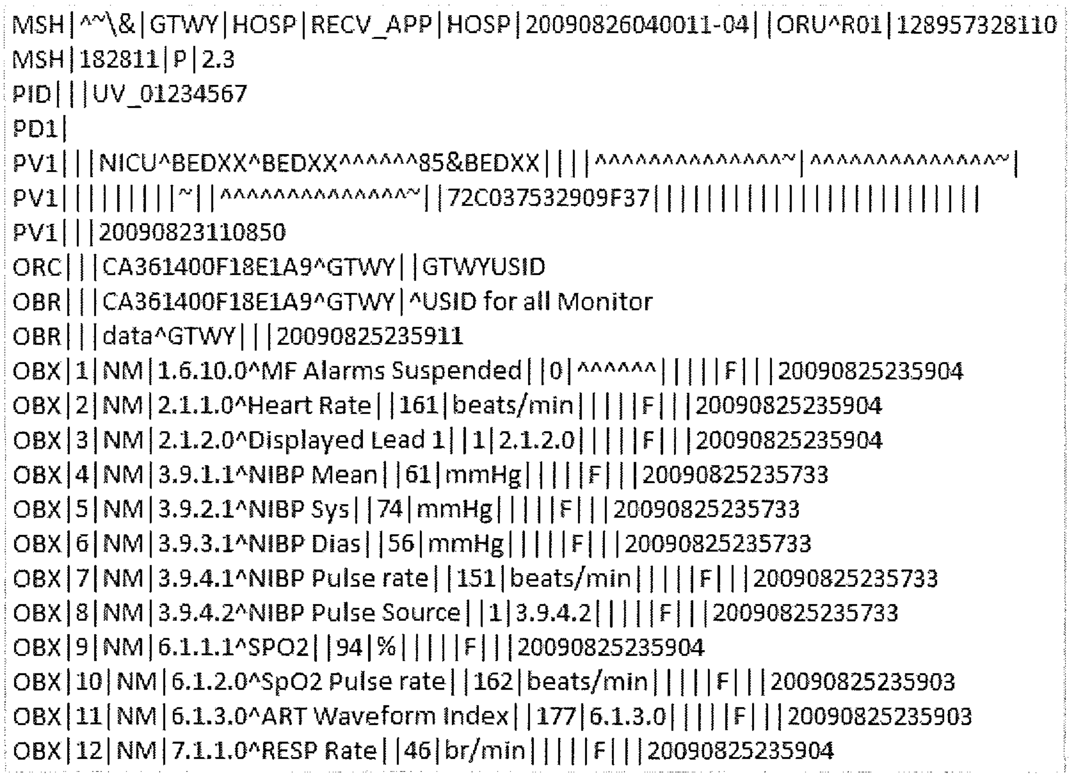
6.1 The Multidimensional Distributed Environment

There are three main distributed sites which will be considered in this scenario. The first deployment is located at The Hospital for Sick Children, Toronto, Ontario (as seen in Figure 6.3). Multiple streams of physiological data are being generated from this location from the Philips IntelliVue MP70 neonatal monitors at the rate of a reading every 1024 milliseconds. These include the constant collection of electrocardiogram derived heart rate (ECG-HR), transcutaneous oxygen saturation (SpO_2) and respiration rate (RR) which is standard clinical practice for all patients in the NICU at The Hospital for Sick Children. Diastolic, systolic and mean blood pressures (DBP, SBP and MBP) are also available when collected as part of clinical practice. Currently, these streams are being used as part of research into earlier detection of late onset neonatal sepsis. To date a combined data set equalling around 115726985 readings has been collected. The complete Artemis deployment occurs in two locations, namely at The Hospital for Sick Children and the UOIT Health Informatics Research (HIR) laboratory and currently supports eight concurrent patients. The following three components are located at The Hospital for Sick Children:

1. The first is responsible for data acquisition from the medical data hub.
2. The second for online analysis utilizing the InfoSphere Streams Runtime from IBM.
3. The third for stream or data persistence utilizing the data integration manager.

Data Persistence occurs to support Online Analysis and Knowledge Extraction. An incremental backup of the data is made each day to a persistence storage mirror located at UOIT and used by the Knowledge Extraction component (as seen in Figure 6.1) at UOIT for knowledge discovery. Redeployment occurs after this step which is where new rules are translated to Stream Processing Application Declarative Engine (SPADE) which is an intermediate language for flexible composition of parallel and distributed data-flow graphs. SPADE allows for potential future deployment in the Online Analysis to monitor future patients in real-time.

The second deployment is situated at The Women and Infants Hospital (WIHRI) in Rhode Island, United States. This site makes use of the SpaceLabs Ultraview SL patient monitors to collect HR, RR, SpO_2, Pulse Rate derived from SpO2 sensor and, where collected, continuous DBP, SBP and MBP. The frequency of data coming from this site is in the form of spot readings taken every minute and stored in its raw form at the UOIT. In order to enable data collection from WIHRI, a cloud based environment is setup where data is transported via a secure tunnel to UOIT in the form of HL7 formatted data packets (as seen in Figure 6.2). In this environment, components of the Data Acquisition exist across both sites and all remaining Artemis components are situated at UOIT instead of the hospital. Presently, the data set from WIHRI amounts to around 3654615 records.



```
MSH|^~\&|GTWY|HOSP|RECV_APP|HOSP|20090826040011-04||ORU^R01|128957328110
MSH|182811|P|2.3
PID|||UV_01234567
PD1|
PV1|||NICU^BEDXX^BEDXX^^^^^^85&BEDXX||||^^^^^^^^^^^^^^^~|^^^^^^^^^^^^^^^~|
PV1|||||||||~||^^^^^^^^^^^^^^^~||72C037532909F37||||||||||||||||||||||||
PV1|||20090823110850
ORC|||CA361400F18E1A9^GTWY||GTWYUSID
OBR|||CA361400F18E1A9^GTWY|^USID for all Monitor
OBR|||data^GTWY|||20090825235911
OBX|1|NM|1.6.10.0^MF Alarms Suspended||0|^^^^^^|||||F|||20090825235904
OBX|2|NM|2.1.1.0^Heart Rate||161|beats/min|||||F|||20090825235904
OBX|3|NM|2.1.2.0^Displayed Lead 1||1|2.1.2.0|||||F|||20090825235904
OBX|4|NM|3.9.1.1^NIBP Mean||61|mmHg|||||F|||20090825235733
OBX|5|NM|3.9.2.1^NIBP Sys||74|mmHg|||||F|||20090825235733
OBX|6|NM|3.9.3.1^NIBP Dias||56|mmHg|||||F|||20090825235733
OBX|7|NM|3.9.4.1^NIBP Pulse rate||151|beats/min|||||F|||20090825235733
OBX|8|NM|3.9.4.2^NIBP Pulse Source||1|3.9.4.2|||||F|||20090825235733
OBX|9|NM|6.1.1.1^SPO2||94|%|||||F|||20090825235904
OBX|10|NM|6.1.2.0^SpO2 Pulse rate||162|beats/min|||||F|||20090825235903
OBX|11|NM|6.1.3.0^ART Waveform Index||177|6.1.3.0|||||F|||20090825235903
OBX|12|NM|7.1.1.0^RESP Rate||46|br/min|||||F|||20090825235904
```

Figure 6.2 - An example of the HL7 data file

The third and final deployment is located at UOIT and comprises of 30 second spot readings of retrospective data from The Hospital for Sick Children collected over a time span of two years. The main purpose of this deployment is to support research for the early detection of multiple clinical diagnoses such as neonatal sepsis and apnoea. As such it contains the Data Persistence, Knowledge Extraction and Redeployment components only.

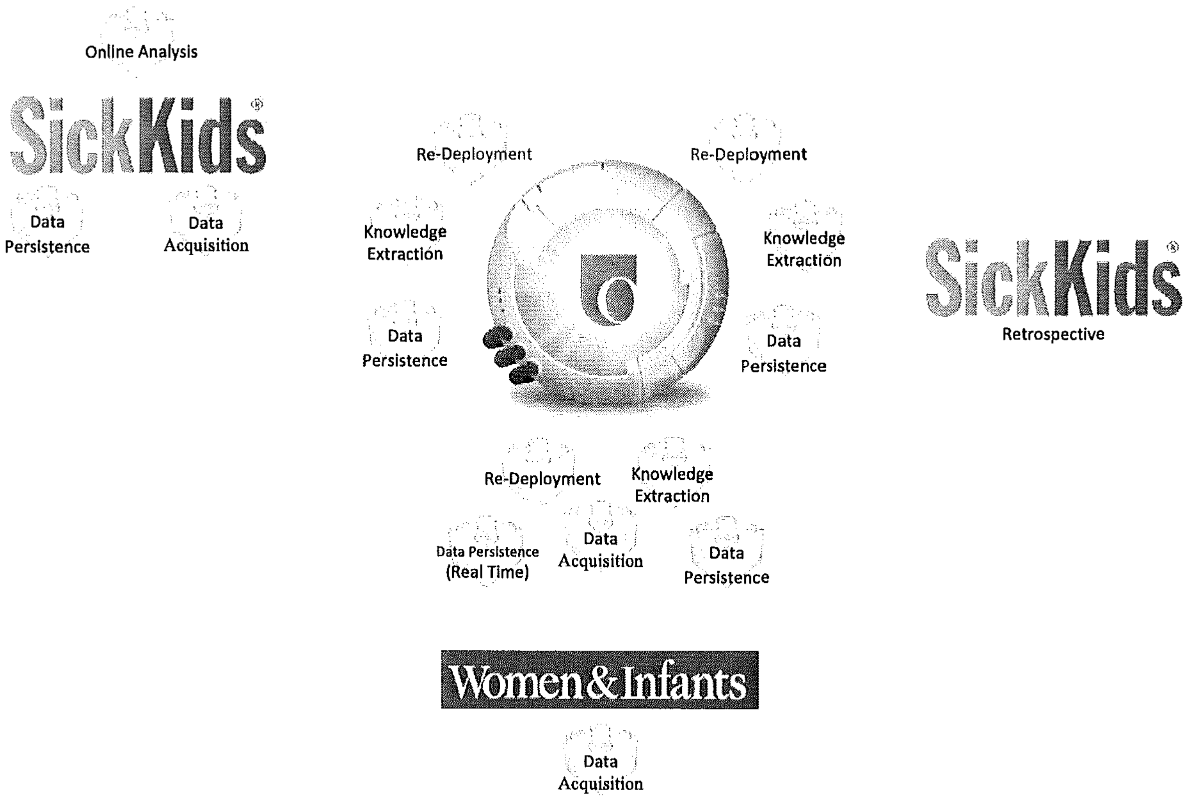


Figure 6.3 - The Multidimensional Distributed Environment

5.1.1 Technical Challenges with Multidimensional Distributed Data

The Multidimensional Distributed Data being collected from the three NICU sites poses some inherent challenges that can prevent normalization of data across the different sites. The main challenge is the differences in data frequency that exists from one location to the next. As highlighted earlier, each site generates data differently which leads to the lack of consistency. For instance:

- The Hospital for Sick Children supplies data at the rate of a reading every 1024 milliseconds

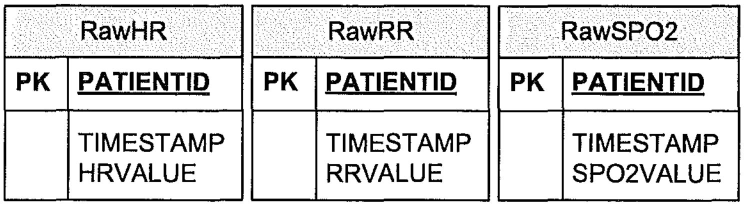
- WIHRI supplies data in the form of spot readings taken every minute
- UOIT – Retrospective Data which comprises of 30 second spot readings.

Normalization of data is the first solution that comes to mind when we discuss the varied frequencies of data collected. However, the data cannot be normalized because different frequencies are required depending on the type of analysis that needs to be performed. For example, trend analysis temporal abstractions on raw heart rate and respiratory rate data could be performed at spot readings taken every 30 seconds. However, this same technique cannot be applied in the analysis of apnoea because apnoea events can occur between two consecutive 30 second spot readings and hence for example transient falls in blood oxygen saturation of less than 30 seconds would be missed. Thus, we need to categorize the abstractions based on similarity as well as frequency in order to effectively run them in a distributed environment.

6.2 Data Structure

The Knowledge Extraction component of Artemis implements the $STDM^n{}_0$ framework. In order to perform temporal abstractions on data, it must first be processed from its raw format. The role of the processing agent is to initiate the collection of stored physiological and clinical data from external data stores supporting the online analysis or collected via some other means outside of Artemis. $STDM^n{}_0$ is the technique used in the knowledge extraction component of Artemis. Within the first two distributed sites as detailed above the processing agent performs the replication of the data from the Online Analysis Data Persistence component to the Knowledge Extraction Data Persistence component. Once the data has passed from the external collection phase, the processing agent converts the data to the required format if and as necessary and then the data is structured and stored in the clinical data and physiological data tables (see

Figure 6.4) accessible by the Knowledge Extraction component. After the completion of this phase, the Temporal Agent begins to process data in order to create the Temporal Abstractions.



RawHR	
PK	**PATIENTID**
	TIMESTAMP HRVALUE

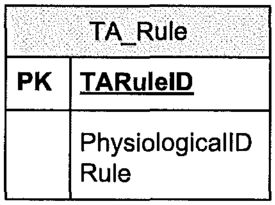
RawRR	
PK	**PATIENTID**
	TIMESTAMP RRVALUE

RawSPO2	
PK	**PATIENTID**
	TIMESTAMP SPO2VALUE

Figure 6.4 - Structure of the tables created by the Processing Agent

6.2.1 The Temporal Phase

Once the processing agent structures and stores the data in local data stores, the Temporal Agent takes over to process the data using rules defined in the Temporal Rules Table (Figure 6.5). Temporal Abstractions are created using the temporal rules and the physiological data that has been collected from the monitoring devices.



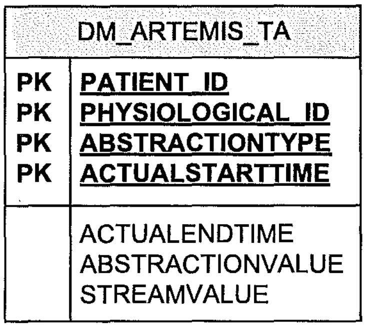
TA_Rule	
PK	**TARuleID**
	PhysiologicalID Rule

Figure 6.5 - Structure of the Temporal Rules Table

As outlined in Chapter 4, there are six main functions of the Temporal Agent:

1. The first step is to retrieve the physiological data from the physiological data store for each parameter for each patient
2. Next, relevant abstraction rules are retrieved from the Temporal Rules Table
3. The rules are then applied to the physiological data, creating simple abstractions for individual data streams for individual patients

4. The created abstractions are then stored in the STDM$^{n}_{0}$ Temporal Data store (Figure 6.6)
5. Complex abstractions are now created from the simple abstractions (from point 3 above), based on any rules found in the Temporal Rules Table
6. Finally, any complex abstractions that are created are then stored in the STDM$^{n}_{0}$ Temporal Data store (Figure 6.6).

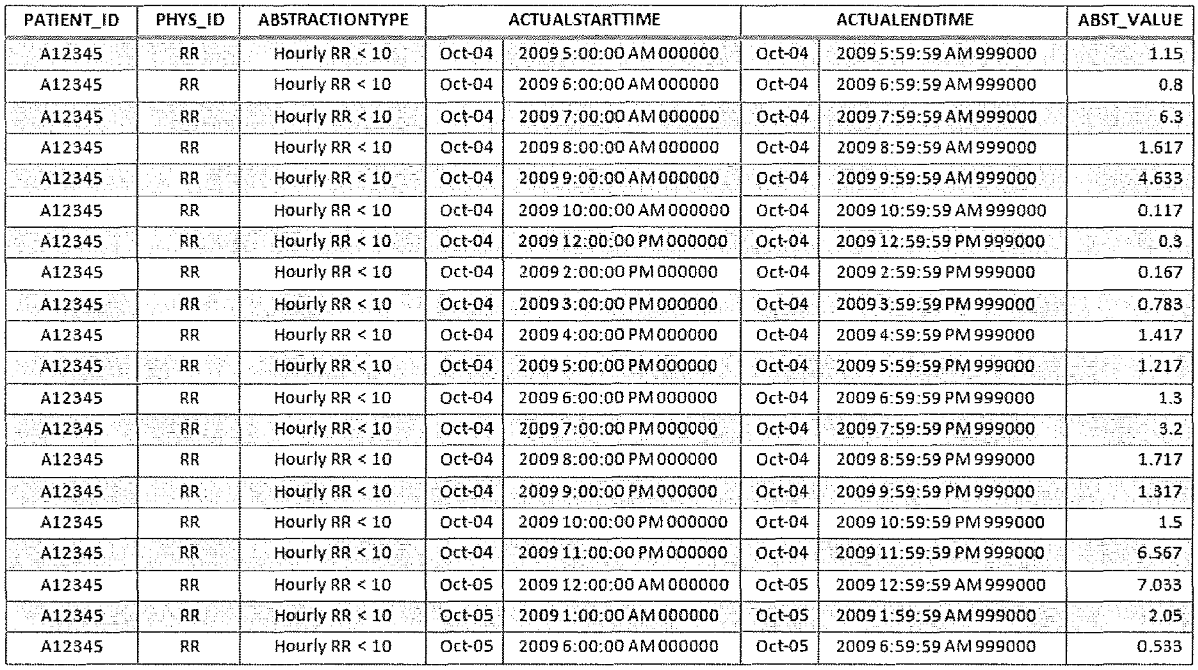


	DM_ARTEMIS_TA
PK	**PATIENT_ID**
PK	**PHYSIOLOGICAL_ID**
PK	**ABSTRACTIONTYPE**
PK	**ACTUALSTARTTIME**
	ACTUALENDTIME ABSTRACTIONVALUE STREAMVALUE

Figure 6.6 - Structure of the Temporal Abstraction Table

Data for each patient may consist of multiple time stamped data streams. The time stamped physiological readings are first abstracted individually to simple temporal abstractions and later can be used to create complex abstractions. A typical abstraction may address level shifts i.e. increase, decrease or stable from point x or trends i.e. changes over a set period. Since a time stamped physiological reading for a certain patient can be part of a number of simple abstractions it is computationally efficient to perform both types of abstractions on one data set.

In order to elaborate further, we take the example of an abstraction run hourly on the respiratory rate (RR) value in a non-distributed setting. In order to analyze patient data, we consider a 60 minute period for our abstraction with the goal of finding when the RR value falls below a specific threshold, which in this case is a value with a threshold of 10. The abstractions created are stored in the Temporal Abstraction Table (as seen the Table 6.1) which condenses and adds context to the data.



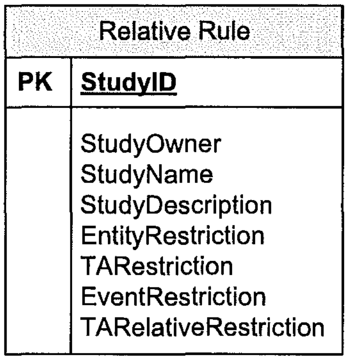
PATIENT_ID	PHYS_ID	ABSTRACTIONTYPE	ACTUALSTARTTIME		ACTUALENDTIME		ABST_VALUE
A12345	RR	Hourly RR < 10	Oct-04	2009 5:00:00 AM 000000	Oct-04	2009 5:59:59 AM 999000	1.15
A12345	RR	Hourly RR < 10	Oct-04	2009 6:00:00 AM 000000	Oct-04	2009 6:59:59 AM 999000	0.8
A12345	RR	Hourly RR < 10	Oct-04	2009 7:00:00 AM 000000	Oct-04	2009 7:59:59 AM 999000	6.3
A12345	RR	Hourly RR < 10	Oct-04	2009 8:00:00 AM 000000	Oct-04	2009 8:59:59 AM 999000	1.617
A12345	RR	Hourly RR < 10	Oct-04	2009 9:00:00 AM 000000	Oct-04	2009 9:59:59 AM 999000	4.633
A12345	RR	Hourly RR < 10	Oct-04	2009 10:00:00 AM 000000	Oct-04	2009 10:59:59 AM 999000	0.117
A12345	RR	Hourly RR < 10	Oct-04	2009 12:00:00 PM 000000	Oct-04	2009 12:59:59 PM 999000	0.3
A12345	RR	Hourly RR < 10	Oct-04	2009 2:00:00 PM 000000	Oct-04	2009 2:59:59 PM 999000	0.167
A12345	RR	Hourly RR < 10	Oct-04	2009 3:00:00 PM 000000	Oct-04	2009 3:59:59 PM 999000	0.783
A12345	RR	Hourly RR < 10	Oct-04	2009 4:00:00 PM 000000	Oct-04	2009 4:59:59 PM 999000	1.417
A12345	RR	Hourly RR < 10	Oct-04	2009 5:00:00 PM 000000	Oct-04	2009 5:59:59 PM 999000	1.217
A12345	RR	Hourly RR < 10	Oct-04	2009 6:00:00 PM 000000	Oct-04	2009 6:59:59 PM 999000	1.3
A12345	RR	Hourly RR < 10	Oct-04	2009 7:00:00 PM 000000	Oct-04	2009 7:59:59 PM 999000	3.2
A12345	RR	Hourly RR < 10	Oct-04	2009 8:00:00 PM 000000	Oct-04	2009 8:59:59 PM 999000	1.717
A12345	RR	Hourly RR < 10	Oct-04	2009 9:00:00 PM 000000	Oct-04	2009 9:59:59 PM 999000	1.317
A12345	RR	Hourly RR < 10	Oct-04	2009 10:00:00 PM 000000	Oct-04	2009 10:59:59 PM 999000	1.5
A12345	RR	Hourly RR < 10	Oct-04	2009 11:00:00 PM 000000	Oct-04	2009 11:59:59 PM 999000	6.567
A12345	RR	Hourly RR < 10	Oct-05	2009 12:00:00 AM 000000	Oct-05	2009 12:59:59 AM 999000	7.033
A12345	RR	Hourly RR < 10	Oct-05	2009 1:00:00 AM 000000	Oct-05	2009 1:59:59 AM 999000	2.05
A12345	RR	Hourly RR < 10	Oct-05	2009 6:00:00 AM 000000	Oct-05	2009 6:59:59 AM 999000	0.533

Table 6.1 - Hourly RR Temporal Abstraction Results

Table 6.1 highlights the output of the Temporal Abstraction (TA) process relating to Respiratory Rate Variability (RRV). Once the TAs are deployed via the TA_Rule table, hourly summaries of RRV are created and stored in the TA table. In this case the event of interest for the TA was the drop in the RR value below 10 within a set period.

6.2.2 The Relative Alignment Phase

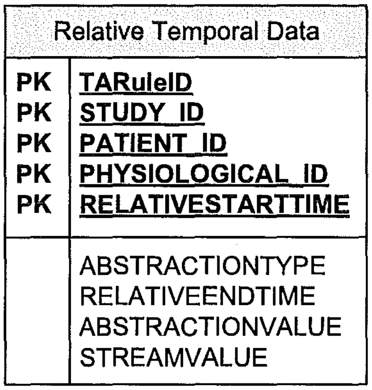
Once the Temporal Agent has created the abstractions from physiological data, it is common for this data to be used in various clinical research studies. Once the abstractions have been created they are stored locally in the STDM$^n{}_0$ data stores until they are needed for a particular study. When a study is prepared, it will often be necessary to realign the time of abstractions relative to a particular point in time of interest. The Relative Rule table (Figure 6.7), which specifies a particular alignment of abstractions for a particular study, holds the information about any relative rules that may need to be applied to the abstractions stored in the Temporal Abstraction table.



Relative Rule	
PK	**StudyID**
	StudyOwner StudyName StudyDescription EntityRestriction TARestriction EventRestriction TARelativeRestriction

Figure 6.7 - Structure of the Relative Rule Table

The Relative Agent realigns the time of abstractions relative to a particular point in time that is of interest by calculating the start and finish times for each abstraction relative to a particular event. If the aim of a research study is to find new trends and patterns that can be indicative of the onset of a condition it will be essential to realign the time of each patient's abstractions relative to the time of the patient being diagnosed with the condition. The abstractions that have been relatively aligned are then stored in the relative temporal data store for further processing. It is also common for different research studies to use the same temporal abstractions which can lead to different re-alignment techniques to be applied to the same data. This is also the reason why every re-aligned Temporal Abstraction is stored in the relative temporal data table (Figure 6.8) specific to the study that has utilized it.

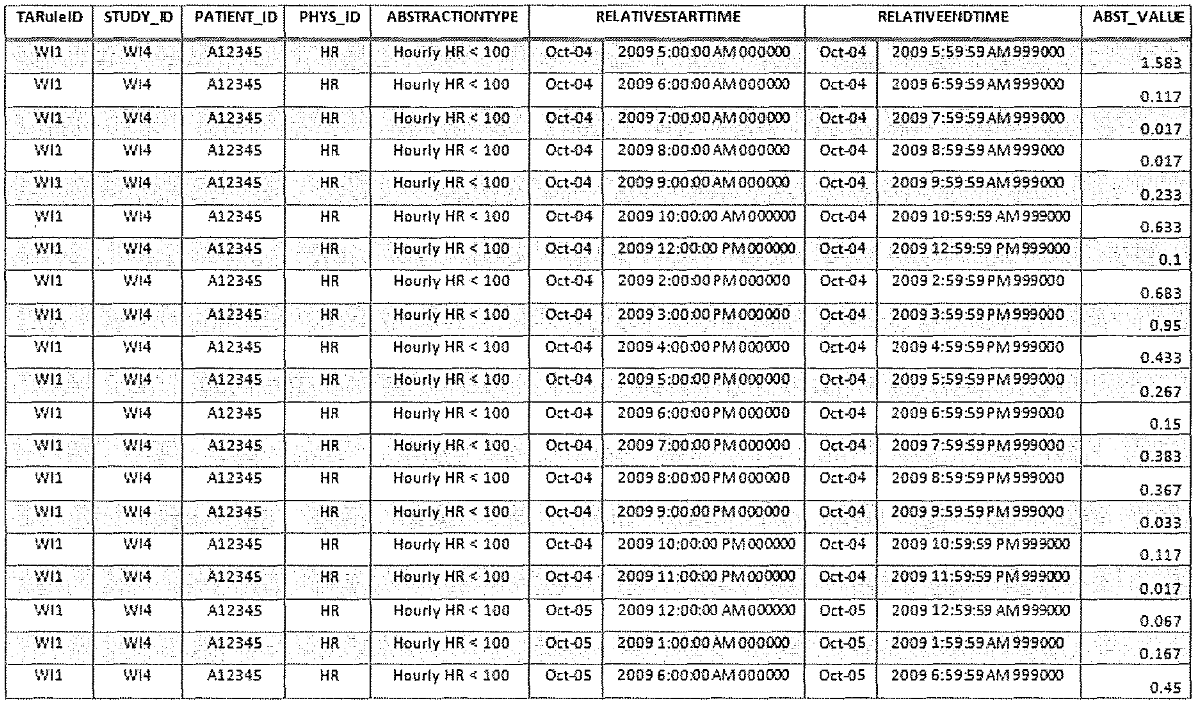


Relative Temporal Data

PK	**TARuleID**
PK	**STUDY_ID**
PK	**PATIENT_ID**
PK	**PHYSIOLOGICAL_ID**
PK	**RELATIVESTARTTIME**
	ABSTRACTIONTYPE RELATIVEENDTIME ABSTRACTIONVALUE STREAMVALUE

Figure 6.8 - Structure of the Relative Temporal Data table

Table 6.2 displays the structure of the Relative Temporal Abstraction table in its current state. In order to analyze patient data, we consider a 60 minute period for our abstraction with the goal of finding when the HR value falls below a specific threshold, which in this case has a threshold value of 100.



TARuleID	STUDY_ID	PATIENT_ID	PHYS_ID	ABSTRACTIONTYPE	RELATIVESTARTTIME		RELATIVEENDTIME		ABST_VALUE
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 5:00:00 AM 000000	Oct-04	2009 5:59:59 AM 999000	1.583
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 6:00:00 AM 000000	Oct-04	2009 6:59:59 AM 999000	0.117
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 7:00:00 AM 000000	Oct-04	2009 7:59:59 AM 999000	0.017
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 8:00:00 AM 000000	Oct-04	2009 8:59:59 AM 999000	0.017
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 9:00:00 AM 000000	Oct-04	2009 9:59:59 AM 999000	0.233
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 10:00:00 AM 000000	Oct-04	2009 10:59:59 AM 999000	0.633
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 12:00:00 PM 000000	Oct-04	2009 12:59:59 PM 999000	0.1
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 2:00:00 PM 000000	Oct-04	2009 2:59:59 PM 999000	0.683
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 3:00:00 PM 000000	Oct-04	2009 3:59:59 PM 999000	0.95
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 4:00:00 PM 000000	Oct-04	2009 4:59:59 PM 999000	0.433
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 5:00:00 PM 000000	Oct-04	2009 5:59:59 PM 999000	0.267
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 6:00:00 PM 000000	Oct-04	2009 6:59:59 PM 999000	0.15
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 7:00:00 PM 000000	Oct-04	2009 7:59:59 PM 999000	0.383
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 8:00:00 PM 000000	Oct-04	2009 8:59:59 PM 999000	0.367
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 9:00:00 PM 000000	Oct-04	2009 9:59:59 PM 999000	0.033
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 10:00:00 PM 000000	Oct-04	2009 10:59:59 PM 999000	0.117
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 11:00:00 PM 000000	Oct-04	2009 11:59:59 PM 999000	0.017
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-05	2009 12:00:00 AM 000000	Oct-05	2009 12:59:59 AM 999000	0.067
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-05	2009 1:00:00 AM 000000	Oct-05	2009 1:59:59 AM 999000	0.167
WI1	WI4	A12345	HR	Hourly HR < 100	Oct-05	2009 6:00:00 AM 000000	Oct-05	2009 6:59:59 AM 999000	0.45

Table 6.2 - HR Relative Temporal Abstraction Results

In order to explain the Relative Temporal Abstraction process further, consider the example of Heart Rate Variability (HRV). Once the TAs have been deployed via the TA_Rule table, hourly summaries of HRV are created and stored in the TA table. In this case the event of interest for the TA was the drop in the HR value below 100 within a set period.

To enable the detection of particular patterns of this abstraction at a particular time before the event of interest, re-alignment of the abstractions relative to the time of the event of interest is necessary. The periods of interest would be abstracted by the temporal agent and stored locally at each site. The role of the relative agent is to re-align the time of the TAs that have been created previously, with an event of interest, thus giving the relative TAs a start time and end time relative to the point of diagnosis. This will enable the comparison and mining of the abstractions to identify particular behaviours that may indicate the onset of the condition being researched.

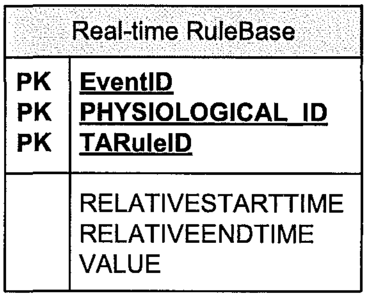
6.2.3 The Functional Agent

The re-aligned Temporal Abstractions form the basis for exploratory and confirmatory data mining, processed by the Functional Agent. The Functional Agent performs data mining tasks used to enable detection of interesting trends and patterns for a particular study. Exploratory data mining is used to analyse the re-aligned Temporal Abstractions across multiple data streams for multiple patients in order to detect new trends and patterns that might present in the data prior to or after the event of interest. The Temporal Abstractions created from the physiological data for each patient that is part of the study must be realigned based on the time of diagnosis as this allows for the search and comparison of all the patients' abstractions regardless of the actual time of the abstractions or the actual time of diagnosis.

Once possible trends and patterns have been discovered, they need to be evaluated by the clinician to enable the creation of a hypothesis. This also allows for the selection of the rules of significance based on the results of the exploratory data mining exercise. The next phase of confirmatory data mining begins with the formulation of the null hypothesis for any results that arouse interest and further investigation.

6.2.4 Rules Generating Agent

The Rules Generating Agent utilizes findings made by the Functional Agent to allow for the creation of rules that can be defined in the real-time rules database (Figure 6.9).



Real-time RuleBase	
PK **PK** **PK**	**EventID** **PHYSIOLOGICAL_ID** **TARuleID**
	RELATIVESTARTTIME RELATIVEENDTIME VALUE

Figure 6.9 - Structure of the Real time RuleBase table

The hypotheses created via the exploratory data mining phase are used by the Rules Generating Agent to create rules that can be stored and utilized by an event stream processor which allows for the application of abstractions on real-time data streams which in turn can help establish these rules in a live analytical system to aid clinicians in real time analysis of data.

6.3 Distribution of Temporal Rules

As discussed in Chapter 4, one of the limitations of the $STDM^n_0$ framework was the notion of only one Temporal Rule table which does not address the area of data distribution and lacks a structure which can support multicenter studies. Another limitation was the lack of clarity on how the Temporal Abstractions will be kept consistent in different sites. In this section, we present a demonstration of the distributed temporal rules environment which highlights the management of multiple Temporal Rule tables (Figure 6.10). This will also enable the Temporal Abstractions to be consistent across the distributed sites.

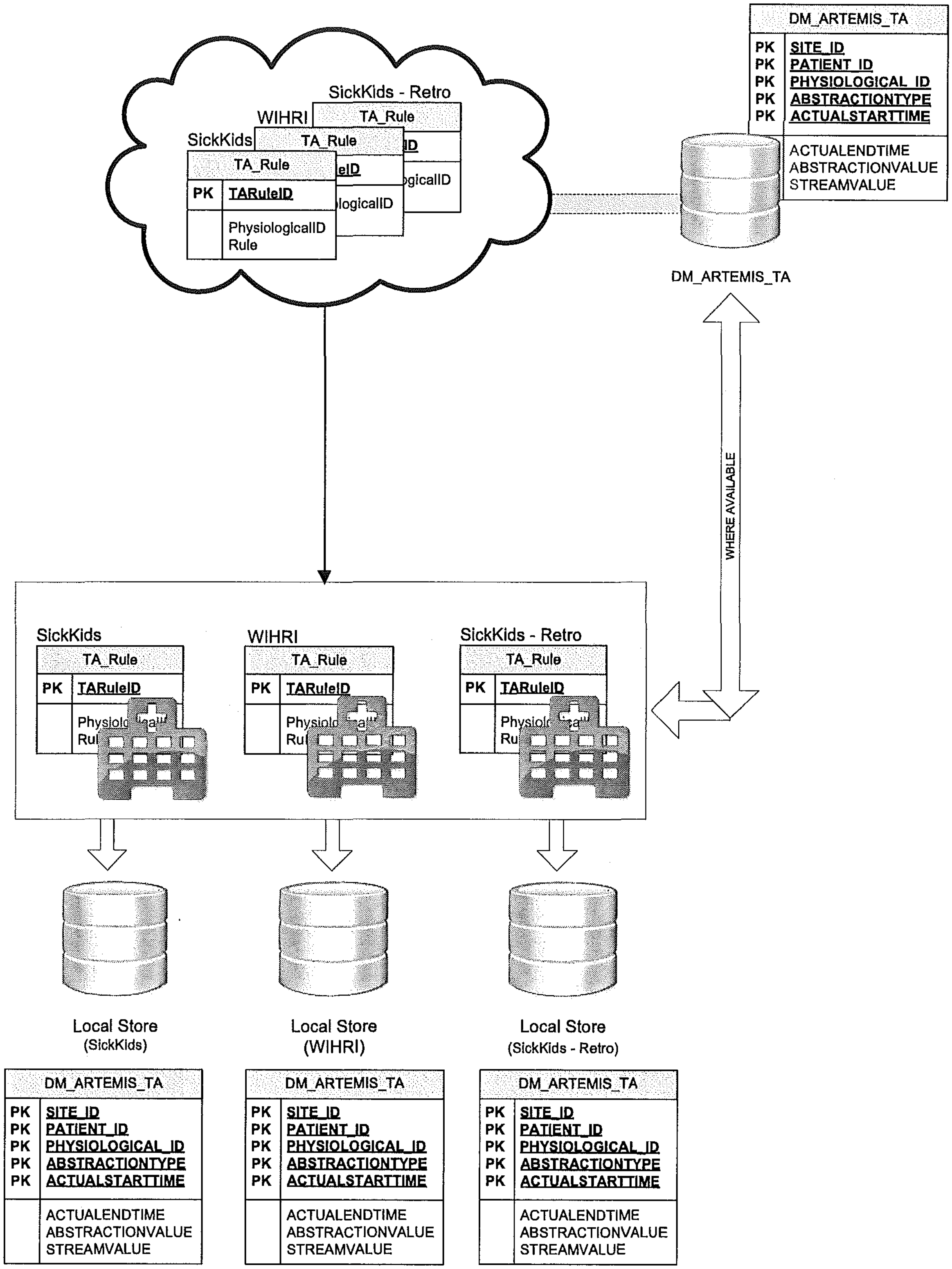
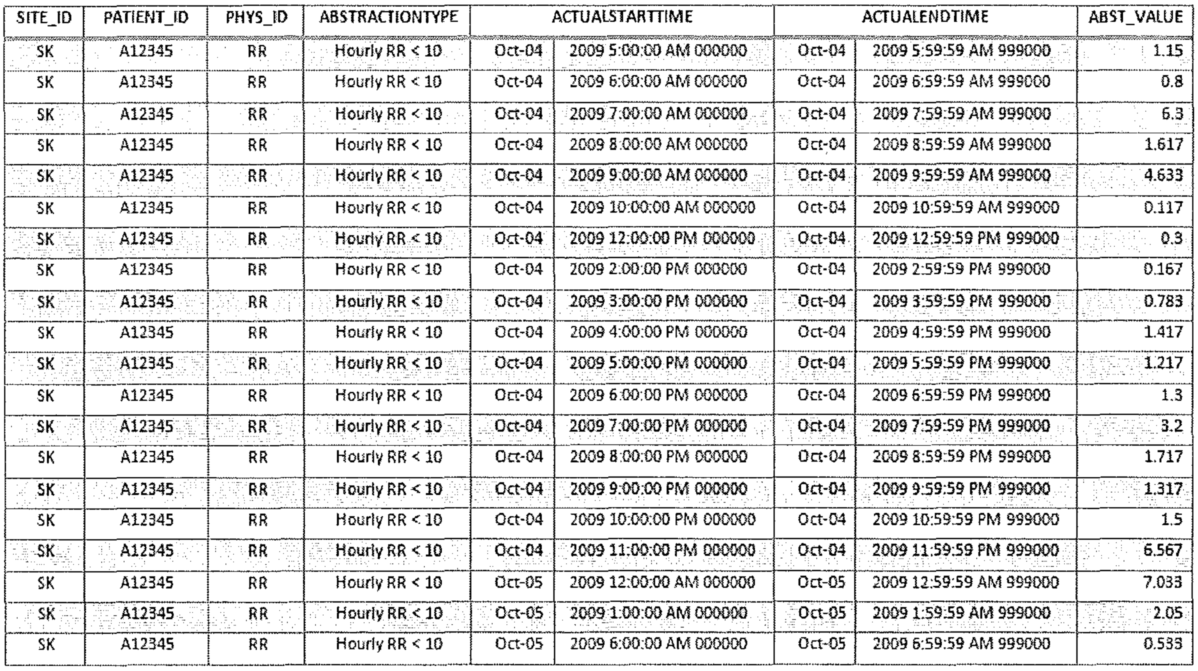


Figure 6.10 - Distribution of Temporal Rules

As discussed, we have three different multidimensional distributed sites which would need to run the Temporal Abstractions. Due to current health care policies and improved patient privacy concerns, it is required that certain types of data exist locally at each site. However, the Temporal Rules do not contain patient identifying information and thus can be decentralized to allow for consistency, better control over the security and better accessibility. In the case of our multidimensional distributed environment, there are four main steps to enable the distribution of some of the data:

1. The Temporal Rules exist at a central hub i.e. at UOIT in this senario. When TA's need to be run, the associated rules are deployed simultaneously for each participating site. The TA rules deployed for each site also contain the SQL query that needs to be run to perform the abstraction at each site as this is supported by the TA_Rule table.
2. Once the Temporal Rules have been deployed, they are run locally at each of the three sites.
3. A SITE_ID tag is also attached to each abstraction that is run at these sites in order to allow for comparison of results across sites when needed (Table 6.2).
4. Finally, the results of the Temporal Abstractions are stored locally at each site (DM_ARTEMIS_TA). Where available, these results will also be populated back at the central UOIT store under the DM_ARTEMIS_TA data table.

Table 6.3 outlines the structure of the distributed Temporal Abstraction tables as they exist at each local multidimensional distributed site. In this table, the data shown contains a SITE_ID tag of SK indicating the data belongs to The Hospital of Sick Children. A similar structure is adopted for each distributed site which is identified by their unique SITE_ID i.e. WIHRI being identified as WI and the SickKids Retrospective data being identified as SK30.



SITE_ID	PATIENT_ID	PHYS_ID	ABSTRACTIONTYPE	ACTUALSTARTTIME		ACTUALENDTIME		ABST_VALUE
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 5:00:00 AM 000000	Oct-04	2009 5:59:59 AM 999000	1.15
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 6:00:00 AM 000000	Oct-04	2009 6:59:59 AM 999000	0.8
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 7:00:00 AM 000000	Oct-04	2009 7:59:59 AM 999000	6.3
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 8:00:00 AM 000000	Oct-04	2009 8:59:59 AM 999000	1.617
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 9:00:00 AM 000000	Oct-04	2009 9:59:59 AM 999000	4.633
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 10:00:00 AM 000000	Oct-04	2009 10:59:59 AM 999000	0.117
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 12:00:00 PM 000000	Oct-04	2009 12:59:59 PM 999000	0.3
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 2:00:00 PM 000000	Oct-04	2009 2:59:59 PM 999000	0.167
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 3:00:00 PM 000000	Oct-04	2009 3:59:59 PM 999000	0.783
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 4:00:00 PM 000000	Oct-04	2009 4:59:59 PM 999000	1.417
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 5:00:00 PM 000000	Oct-04	2009 5:59:59 PM 999000	1.217
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 6:00:00 PM 000000	Oct-04	2009 6:59:59 PM 999000	1.3
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 7:00:00 PM 000000	Oct-04	2009 7:59:59 PM 999000	3.2
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 8:00:00 PM 000000	Oct-04	2009 8:59:59 PM 999000	1.717
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 9:00:00 PM 000000	Oct-04	2009 9:59:59 PM 999000	1.317
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 10:00:00 PM 000000	Oct-04	2009 10:59:59 PM 999000	1.5
SK	A12345	RR	Hourly RR < 10	Oct-04	2009 11:00:00 PM 000000	Oct-04	2009 11:59:59 PM 999000	6.567
SK	A12345	RR	Hourly RR < 10	Oct-05	2009 12:00:00 AM 000000	Oct-05	2009 12:59:59 AM 999000	7.033
SK	A12345	RR	Hourly RR < 10	Oct-05	2009 1:00:00 AM 000000	Oct-05	2009 1:59:59 AM 999000	2.05
SK	A12345	RR	Hourly RR < 10	Oct-05	2009 6:00:00 AM 000000	Oct-05	2009 6:59:59 AM 999000	0.533

Table 6.3 - Structure of the Distributed DM_ARTEMIS_TA Tables

Policies regarding the handling of data and its privacy will always differ across the multidimensional distributed sites; hence the need to support the data in a distributed setting. By having regulatory requirements that will govern where the data has to reside and how it can be interacted with we can manage sensitive patient data properly and at the same time improve patient outcomes at the health facilities.

6.4 Distribution of Relative Rules

Chapter 4 presented details on The Relative Agent which realigns the time of abstractions relative to a particular point in time that is of interest. Depending on the study taking place, the temporal abstractions may need to be realigned relative to a particular point in time if the behaviour of certain parameters in the time leading up to a diagnosis needs to be studied. The Relative Rule table, which specify a particular alignment of abstractions for a particular study, holds the information about any relative rules that may need to be applied to the abstractions stored in the Temporal Abstraction table.

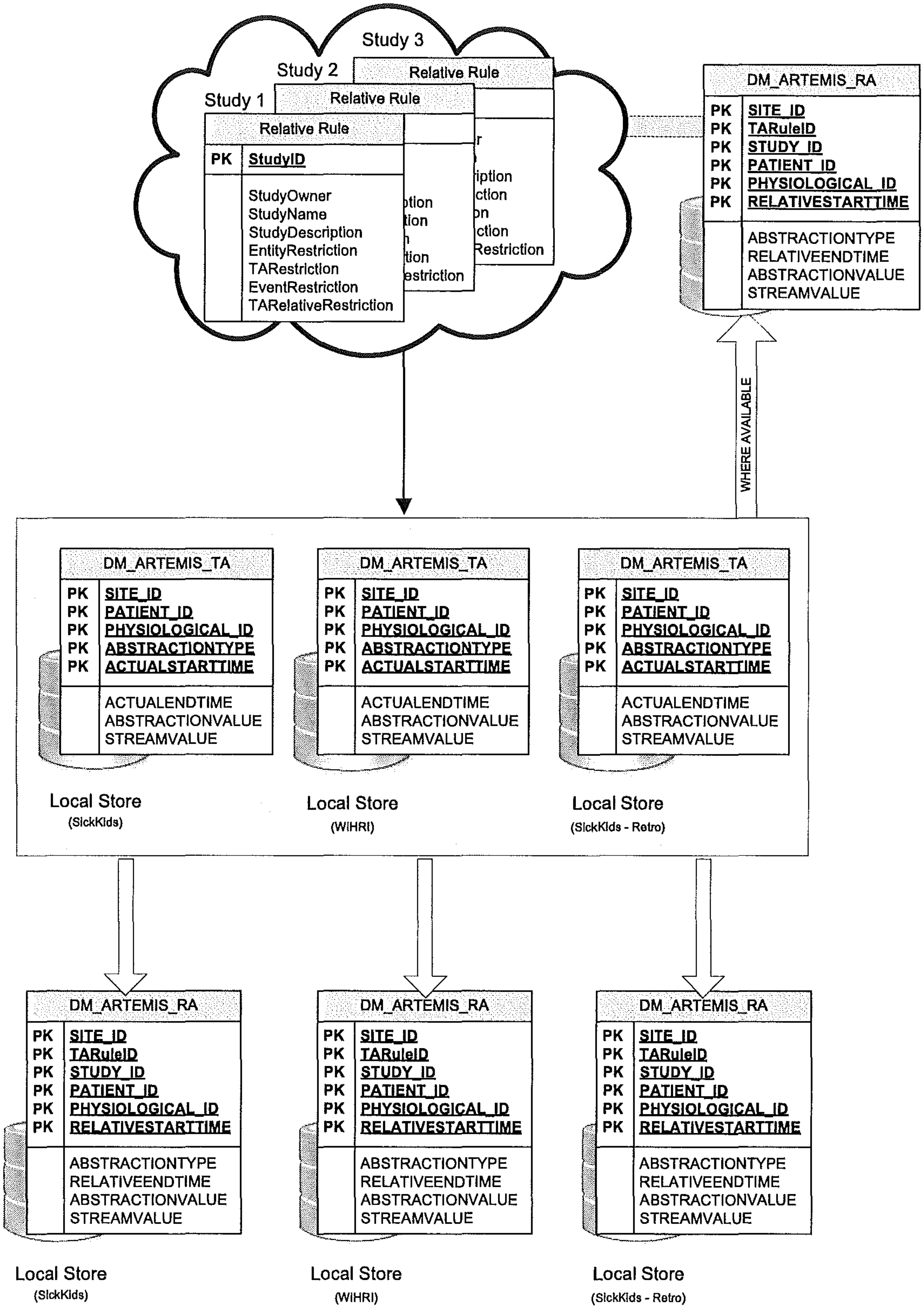
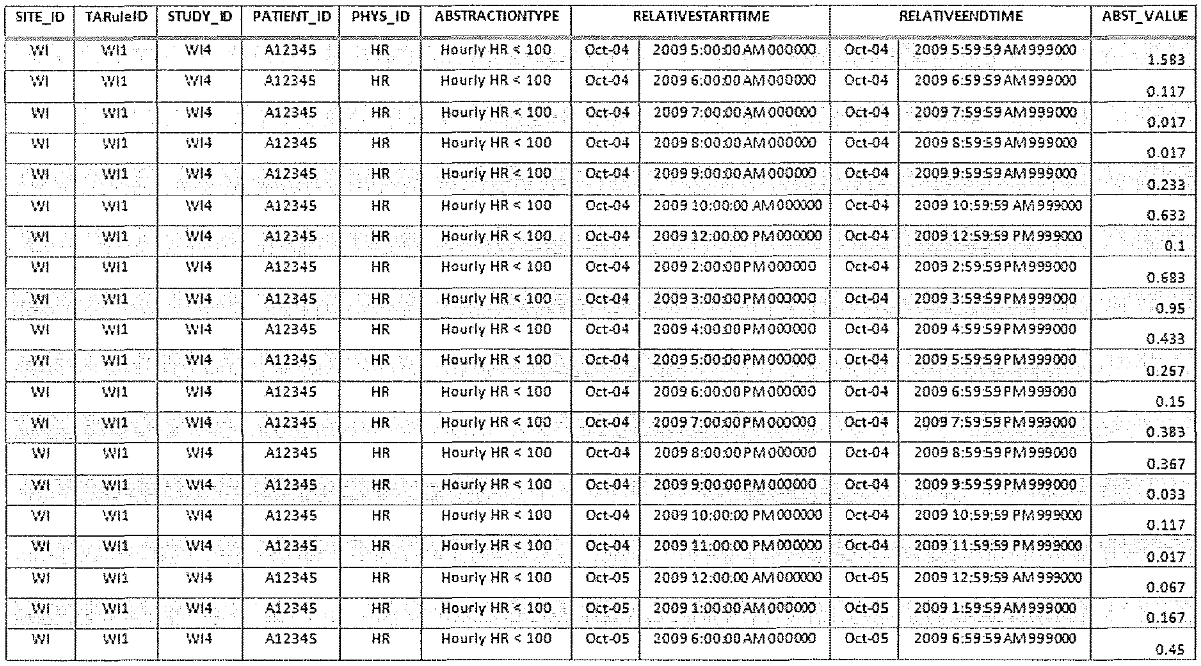


Figure 6.11 - Distribution of Relative Rules

Figure 6.11 outlines the structure supporting the distribution of Relative Rules. As discussed earlier, the STDM$^{n}{}_{0}$ framework presents the notion of only one Relative Rule table which is not suited in multi-centered studies. The following three step approach is taken to enable the distribution of Relative Rules:

1. Relative rules for each study are deployed from the central data store (at UOIT). A separate study table exists for each participating facility and is assigned a unique StudyID
2. Once deployed, the Temporal Abstractions table created at each site is accessed locally in order to perform the Relative Alignments needed for the particular study
3. The re-aligned Temporal Abstractions are then stored in the relative temporal data tables specific to the study and the site. Each site is identified by a unique StudyID and SITE_ID (Table 6.4).
4. Where available, these results will also be populated back at the central UOIT store under the DM_ARTEMIS_RA data table.

Table 6.4 shows an example of the distributed Relative Temporal Abstraction table. The data shown contains a SITE_ID tag of WI indicating the data belongs to The Women's and Infants Hospital. The corresponding TARuleID and unique STUDY_ID attributes are also contained in this table. A similar structure is adopted for each distributed site which is identified by their unique SITE_ID i.e. SickKids being identified as SK and the SickKids Retrospective data being identified as SK30.



SITE_ID	TARuleID	STUDY_ID	PATIENT_ID	PHYS_ID	ABSTRACTIONTYPE	RELATIVESTARTTIME		RELATIVEENDTIME		ABST_VALUE
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 5:00:00 AM 000000	Oct-04	2009 5:59:59 AM 999000	1.583
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 6:00:00 AM 000000	Oct-04	2009 6:59:59 AM 999000	0.117
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 7:00:00 AM 000000	Oct-04	2009 7:59:59 AM 999000	0.017
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 8:00:00 AM 000000	Oct-04	2009 8:59:59 AM 999000	0.017
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 9:00:00 AM 000000	Oct-04	2009 9:59:59 AM 999000	0.233
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 10:00:00 AM 000000	Oct-04	2009 10:59:59 AM 999000	0.633
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 12:00:00 PM 000000	Oct-04	2009 12:59:59 PM 999000	0.1
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 2:00:00 PM 000000	Oct-04	2009 2:59:59 PM 999000	0.683
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 3:00:00 PM 000000	Oct-04	2009 3:59:59 PM 999000	0.95
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 4:00:00 PM 000000	Oct-04	2009 4:59:59 PM 999000	0.433
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 5:00:00 PM 000000	Oct-04	2009 5:59:59 PM 999000	0.257
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 6:00:00 PM 000000	Oct-04	2009 6:59:59 PM 999000	0.15
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 7:00:00 PM 000000	Oct-04	2009 7:59:59 PM 999000	0.383
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 8:00:00 PM 000000	Oct-04	2009 8:59:59 PM 999000	0.367
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 9:00:00 PM 000000	Oct-04	2009 9:59:59 PM 999000	0.033
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 10:00:00 PM 000000	Oct-04	2009 10:59:59 PM 999000	0.117
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-04	2009 11:00:00 PM 000000	Oct-04	2009 11:59:59 PM 999000	0.017
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-05	2009 12:00:00 AM 000000	Oct-05	2009 12:59:59 AM 999000	0.067
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-05	2009 1:00:00 AM 000000	Oct-05	2009 1:59:59 AM 999000	0.167
WI	WI1	WI4	A12345	HR	Hourly HR < 100	Oct-05	2009 6:00:00 AM 000000	Oct-05	2009 6:59:59 AM 999000	0.45

Table 6.4 - Structure of the Distributed DM_ARTEMIS_RA Tables

6.5 Distribution of Rule Base Data

The realigned temporal abstractions created by the Relative Agent are further processed by the Functional Agent. In the STDM$^n_0$ framework the Functional Agent is responsible for data mining tasks used to enable detection of interesting trends and patterns for a particular study. If the particular study is exploring the possibility of communal patterns or trends being exhibited in the physiological data of neonates in the time period leading up to diagnosis of a particular condition, then the Temporal Abstractions created for each patient that is part of the study must be realigned based on the time of diagnosis. This enables the comparison of all the abstractions for all the patients regardless of the actual time of the abstractions and diagnosis.

The Functional Agent utilizes exploratory data mining to detect new trends and patterns in multiple parameters. These trends and patterns are then evaluated by the clinician or researcher to create a hypothesis. Once the hypothesis is created from the result of the exploratory data mining, a null hypothesis can be established and tested with confirmatory data mining techniques.

The Rules Generating Agent processes the hypotheses created by the functional agent into appropriate rules that can be stored in the Rule Base. These rules can further be used in a real-time monitoring system aiding clinicians in the early detection of events of interest for better diagnosis and treatment (as seen in Figure 6.12). The rules coming back in the multicentre studies are used in an iterative way to derive one rule that is applicable across all studies. This is done using null hypothesis testing which allows us to refine the Temporal Abstractions of importance and derive globally applicable rules.

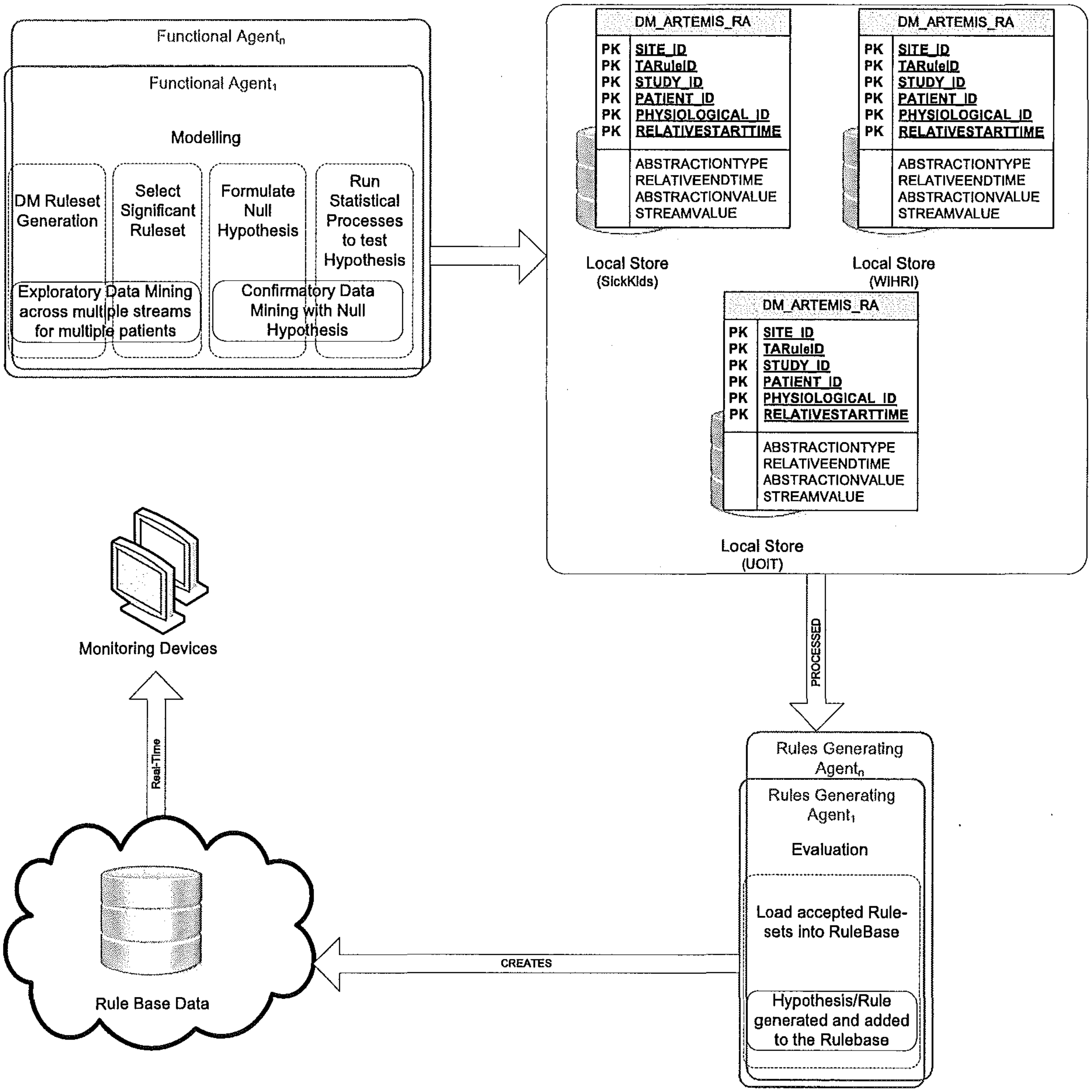


Figure 6.12 - Generation of Rule Base Data

6.6 Conclusion

This chapter presented the application of the S***D***TDM$^{n}_{0}$ framework within the NICU context. The multidimensional distributed environment is presented and the technical challenges associated with the distribution of the data have also been discussed. The data structure at each of the distributed sites has also been presented in this chapter along with a presentation of how the data changes as it processed by the Agents in the STDM$^{n}_{0}$ framework. The distribution of Temporal Rules, Relative Rules and Rule Base data are presented as they would appear in a multidimensional distributed environment along with a description of which components are located centrally and which exist locally.

7. Chapter 7 – Conclusion

7.1 Summary

This thesis presented a service based multidimensional distributed temporal data mining framework which extended the functionality of the existing non-distributed framework. The research is demonstrated through a case study utilising NICU patient physiological time series data streams from three distributed sites.

To highlight the demand for a framework which can support a multidimensional distributed environment, current literature in the area of Distributed Data Mining, Distributed Data Mining of Time Series Data, Temporal Abstraction and Distributed Temporal Abstraction was reviewed in chapter 2. During the literature review no direct reference to the utilization of distributed data mining of multiple and multi-frequency data streams was found; this is a significant limitation because the inherently distributed nature of health care could benefit immensely from distributed data mining applications. The review also revealed several shortfalls in existing systems: a focus on single site rather than several distributed sites; lack of discussion on managing data privacy and confidentiality; a lack of interaction with real time data streams in a distributed environment; and a lack of the concept of synchronous deployment of temporal abstractions.

Resulting from the review, the research hypotheses of this thesis were that:

1. A multidimensional distributed data mining framework can be defined for time series data research for the discovery of trends and patterns prior to a given clinical event.

2. The framework will utilize elements of data fusion and agent-based analysis so that it will work with relational databases and large scale data mining applications.
3. A set of data mining tools can be applied for temporal abstraction, relative alignment and cluster analysis in a distributed manner to support multiple research studies.
4. The framework can be applied in a broad neonatal context addressing issues of data privacy and confidentiality and being deployable as part of multicenter studies while maintaining data integrity at each participating site.

Chapter 3 introduced the Neonatal Intensive Care Unit (NICU) which is the application domain for this research and extended the understanding on the data rich environment of the NICU. This chapter also introduced the clinical distributed problem as well as the implications for this research. As this research extends previous research, chapter 4 introduced the context for this extension by describing the existing architecture of the Service Based Multidimensional Temporal Data Mining (STDM$^{n}_{0}$) framework. The chapter highlighted the need for its operation in a distributed setting and discussed current limitations that make distributed deployment impossible. Chapter 5 addresses the highlighted limitations by introducing the Service Based Multidimensional Distributed Temporal Data Mining (S***D***TDM$^{n}_{0}$) framework. The framework provides a structure to support multicenter studies and allows for the management of the Temporal and Relative Rule tables in a distributed environment, while keeping them consistent across the distributed sites. In chapter 6 the functions of the S***D***TDM$^{n}_{0}$ framework were demonstrated and explained within the context of the NICU.

The research hypotheses have been addressed by this thesis and this is summarised below:

1. *A multidimensional distributed data mining framework can be defined for time series data research for the discovery of trends and patterns prior to a given clinical event.* Chapters 5 and 6 discuss the design and application of the framework in a multidimensional distributed setting.

2. *The framework will utilize elements of data fusion and agent-based analysis so that it will work with relational databases and large scale data mining applications.* This is demonstrated in chapter 6 where a detailed account of the agent based analysis is highlighted.

3. *A set of data mining tools can be applied for temporal abstraction, relative alignment and cluster analysis in a distributed manner to support multiple research studies.* Chapter 5 provides a framework that can support multiple research studies and this is demonstrated further in chapter 6.

4. *The framework can be applied in a broad neonatal context addressing issues of data privacy and confidentiality and being deployable as part of multicenter studies while maintaining data integrity at each participating site.* Chapter 3 provides background into the neonatal context and provides the understanding on the data rich environment of the NICU. Chapter 6 discusses the use of the S***D***TDM$^n{}_0$ framework in a neonatal context, by illustrating the framework's use with real life neonatal monitoring data.

7.2 Contributions

The areas of research contribution to knowledge resulting from this thesis are:

- Extensions to the STDM$^n_0$ framework to allow for the application of the framework in a multidimensional distributed setting.
- Enable the deployment of Temporal and Relative Rules from a distributed setting.
- Enable the synchronization of Temporal and Relative Temporal Data at each of the multidimensional distributed sites.
- Ability to support multiple research studies and a structure allowing for the comparison of results from each study.
- Enabling the distribution of Rule Base Data allowing for the synchronous deployment of Real-Time Rules at each participating site.

7.3 Future Research

Currently, the S***D***TDM$^n_0$ framework is designed to distribute the Temporal Rules, Relative Rules and Rule Base Data but the storage of the Temporal Abstractions and Relative Temporal Abstractions are still local to each site. The S***D***TDM$^n_0$ framework hints towards the possibility to store this data in a cloud environment (Figure 7.1).

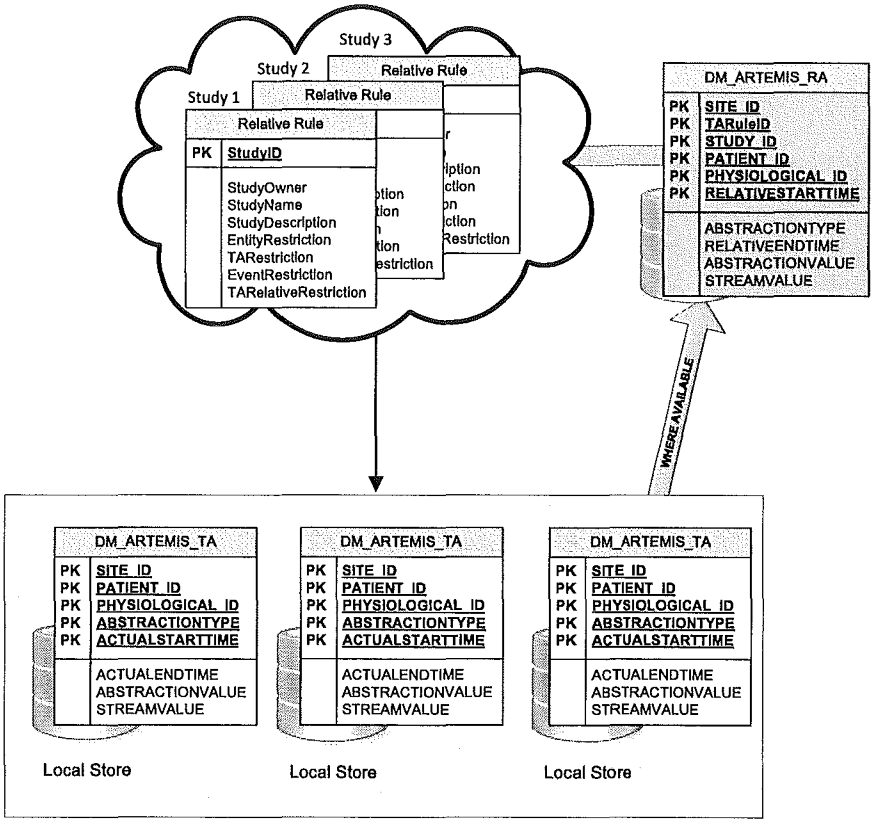


Figure 7. - Potential Cloud Storage of Relative Temporal Abstraction Data

Storage of Temporal Abstractions and Relative Temporal Abstractions has been kept local to each site due to current health care policies and improved patient privacy concerns. However, the creation of regionalized cloud environments can be a potential solution to distribute the Temporal and Relative Temporal Abstractions. For example, each region, province, state or country can have one dedicated cloud environment which can store these abstractions in accordance to the privacy policies governing the particular area. This may also allow for cross site comparison of results in multicenter studies in order to identify trends that may occur globally or only at certain facilities.

Apart from the potential regionalized cloud storage environment of the S***D***TDM$^{n}_{0}$ framework, there are several other opportunities to explore in future work. Firstly, there is an opportunity to test this approach further through clinical research into late onset neonatal sepsis. As mentioned earlier, neonatal sepsis is a common nosocomial infection that affects neonates and has been shown to exhibit changes in physiological data before the condition can be diagnosed through blood cultures. There also lies an opportunity for the testing of the distributed multidimensional data mining technique with other conditions such as Apnoea and Intraventricular Haemorrhage which were highlighted in chapter 3 to be conditions that may greatly benefit from the discoveries made by a distributed data mining framework. These would also include testing for sensitivity and specificity of what is researched in order to confirm the findings and highlight the rules of significance.

Future work will also include further details of the web services for the communications between the distributed sites as well as the implementation of a backup mechanism for the deployment of Temporal and Relative Rules in case the communication link to the cloud is interrupted. This is an important consideration as the connection between a cloud environment and a distributed site is easily influenced by external elements governing each location. Having a strong backend design will ensure synchronous deployment of rules across each site; thus maintaining data consistency. Finally, there is also potential in the application and extension of this work outside the medical domain in areas such as peer-to-peer networking, distributed data mining in mobile environments, stock prediction, fraud prevention and intelligent user interfaces.

7.4 Conclusion

This thesis has presented a framework for clinical research in neonatal intensive care physiological monitoring data by the design of the S***D***TDM$^n{}_0$ framework, a multidimensional distributed data mining framework supporting time series data analysis. A demonstration of the distribution of Temporal and Relative Rules in a multidimensional environment is provided in order to support multicenter studies. A potential area of future work has been discussed to further extend this framework which will allow for greater insights into the abstractions and allow for the comparison of results on a global scale in turn improving the discovery of new knowledge. As the rate of preterm birth and mortality around the world increases, so does the demand for faster diagnosis and quicker treatment of patients. Collaboration between leading health facilities around the world can prove to be a key factor in discovering novel trends and patterns from patient data and consequently improve patient care with faster and more accurate patient diagnosis. The S***D***TDM$^n{}_0$ framework provides a multidimensional distributed data mining structure to collaborating facilities while maintaining consistency of data across the distributed sites and supporting multicenter studies to achieve new strides towards better patient care.

References

Abe, H., & Yamaguchi, T. (2005). *Implementing an integrated time-series data mining environment-a case study of medical kdd on chronic hepatitis.* Data Mining Workshops, 2007. ICDM Workshops 2007. Seventh IEEE International Conference.

Beck S, Wojdyla D, Say L, Betran AP, Merialdi M, Requejo JH, Rubens C, Menon R, Van Look P. 2009. WHO systematic review on maternal mortality and morbidity: The global burden of preterm birth. The Bulletin of the World Health Organization.

Bjering, H., & McGregor, C. (2010). *A multidimensional temporal abstractive data mining framework.* HIKM '10 Proceedings of the Fourth Australasian Workshop on Health Informatics and Knowledge Management (108).

Blount, M., Ebling, M. R., Eklund, J. M., James, A. G., McGregor, C., Percival, N., Sow, D. (2010). Real-time analysis for intensive care: Development and deployment of the artemis analytic system. *Engineering in Medicine and Biology Magazine, IEEE, 29*(2), 110-118.

Boaz, D., & Shahar, Y. (2005). A framework for distributed mediation of temporal-abstraction queries to clinical databases. *Artificial Intelligence in Medicine, 34*(1), 3-24.

Canadian Institute for Health Information (CIHI), Too Early, Too Small: A Profile of Small Babies Across Canada (Ottawa, Ont.: CIHI, 2009).

Canadian Paediatric Society (CPS) (2006). Levels of neonatal care. *Paediatr Child Health, 11*(5). Retrieved from http://www.cps.ca/english/statements/FN/FN06-02.pdf

Catley C., McGregor C., Bjering H. Multi-Dimensional Temporal Data Mining of Time-Series Clinical Data: Survey and Modeling. In Faculty of Business and IT, UOIT.

Catley, C., Smith, K., McGregor, C., James, A., & Eklund, J. (2010). *A framework to model and translate clinical rules to support complex real-time analysis of physiological and clinical data*. IHI '10 Proceedings of the 1st ACM International Health Informatics Symposium.

Constantinescu, Rasvan. (2005). Constructive reaserch diagram. (2005).

Dolin, R. (1995). Modeling the temporal complexities of symptoms. *Journal of the American Medical Informatics Association, 2*(5), 323.

Ferber, J. (1999). *Multi-agent systems: An introduction to distributed artificial intelligence* (Vol. 222): Addison-Wesley New York.

Foster, D. and C. McGregor (2005). Overview of an Agent-based IDSS Framework for Neonatal Analysis and Trend Detection. National Health Informatics Conference (13th : 2005 : Melbourne, Vic.). Brunswick East, Vic., Health Informatics Society of Australia.

Foster, D., & McGregor, C. (2006). *Design of an agent server for neonatal analysis and trend detection*. Interational Transactions on Systems Science and Applications 1(1)

Fu, Y. (2001). Distributed data mining: An overview. *IEEE TCDP newsletter*.

Griffin, M., Lake, D., O'Shea, T., & Moorman, J. (2007). Heart rate characteristics and clinical signs in neonatal sepsis. *Pediatric research, 61*(2), 222.

Hilberman, M. (1975). The evolution of intensive care units. *Critical care medicine, 3*(4).

Hoballah, I., & Varshney, P. (2002). Distributed bayesian signal detection. *Information Theory, IEEE Transactions on, 35*(5), 995-1000.

Khashan, A., McNamee, R., Abel, K., Mortensen, P., Kenny, L., Pedersen, M., . . . Baker, P. (2008). Rates of preterm birth following antenatal maternal exposure to severe life events: A population-based cohort study. *Human Reproduction*.

Kramer, M., Liu, S., Luo, Z., Yuan, H., Platt, R., & Joseph, K. (2002). Analysis of perinatal mortality and its components: Time for a change? *American Journal of Epidemiology, 156*(6).

Martin, J., Hamilton, B., Sutton, P., Ventura, S., Menacker, F., Kirmeyer, S., & Mathews, T. (2009). Births: Final data for 2006. *Public Health Resources*, 65.

Martin-Sanchez, F., Maojo, V., & Lopez-Campos, G. (2002). Integrating genomics into health information systems. *Methods of information in medicine, 41*(1), 25-30.

McGregor, C. (2005). E-baby web services to support local and remote neonatal intensive care. *HIC 2005 and HINZ 2005: Proceedings*, 344.

McGregor, C., & Stacey, M. (2007). High frequency distributed data stream event correlation to improve neonatal clinical management. DEBS '07 Proceedings of the 2007 inaugural international conference on Distributed event-based systems, 19-20.

McGregor, C., & Eklund, J. (2008). *Real-time service-oriented architectures to support remote critical care: Trends and challenges*. Computer Software and Applications, 2008. COMPSAC '08. 32nd Annual IEEE International.

McGregor, C. P. (July 2010). *Patent No. 089705-0009.* Canada, Gatineau Quebec.

Miller, G. (1956). Selection 7 the magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychol. Rev, 63*, 81-97.

Moskovitch, R., Stopel, D., Verduijn, M., Peek, N., De Jonge, E., & Shahar, Y. Analysis of ICU Patients Using the Time Series Knowledge Mining Method. *Journal of Intelligent Information Systems, 12*(1)

Neonatology on the web (2002). Retrieved Jan, 20th, 2011, from http://www.neonatology.org/tour/equipment.html

Nguyen, J., Shahar, Y., Tu, S., Das, A., & Musen, M. (1999). Integration of temporal reasoning and temporal-data maintenance into a reusable database mediator to answer abstract, time-oriented queries: The tzolkin system. *Journal of Intelligent Information Systems, 13*(1), 121-145.

O Connor, M., Grosso, W., Tu, S., & Musen, M. (2001). Rasta: A distributed temporal abstraction system to facilitate knowledge-driven monitoring of clinical databases. *Studies in Health Technology and Informatics*, 508-512.

PHAC, Initials. (2000, 16 06). *Preterm birth.* Retrieved from http://www.phac-aspc.gc.ca/publicat/meas-haut/mu_d_e.html

Portinale, L., Montani, S., Bottrighi, A., Leonardi, G., & Juarez, J. (2006). *A case-based architecture for temporal abstraction configuration and processing. Tools with Artificial Intelligence, IEEE 18 (1)*: 667 - 676

Shahar Y. and Musen MA. RÉSUMÉ: a Temporal-Abstraction System for Patient Monitoring. Computers and Biomedical Research 1993:26: 255-273.

Shahar, Y. (1997). Context-sensitive temporal abstraction of clinical data. *KLUWER INTERNATIONAL SERIES IN ENGINEERING AND COMPUTER SCIENCE*, 37-60.

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

Stark, A. (2004). Levels of neonatal care. *Pediatrics, 114*(5), 1341.

Stacey, M., C. McGregor, et al. (2007). An Architecture for Multi-Dimensional Temporal Abstraction and its Application to Support Neonatal Intensive Care, Health Informatics Research, University of Western Sydney.

Stolfo, S., Prodromidis, A., Tselepis, S., Lee, W., Fan, D., & Chan, P. (1997). *Jam: Java agents for meta-learning over distributed databases. Proceedings of the 3rd ACM SIGMOD International Workshop on Data Mining and Knowledge Discovery ACM Press*, 74–81.

Stonebraker, M., Aoki, P., Litwin, W., Pfeffer, A., Sah, A., Sidell, J., . . . Yu, A. (1996). Mariposa: A wide-area distributed database system. *The VLDB Journal—The International Journal on Very Large Data Bases, 5*(1), 048-063.

Vinoski, S. (2002). Corba: Integrating diverse applications within distributed heterogeneous environments. *Communications Magazine, IEEE, 35*(2), 46-55.

Viswanathan, R., & Varshney, P. (2002). Distributed detection with multiple sensors i. Fundamentals. *Proceedings of the IEEE, 85*(1), 54-63.