Expert Systems with Applications 41 (2014) 7914–7926

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An ontology-based framework to support intelligent data analysis of sensor measurements

Fernando Roda [a], Estanislao Musulin [a,b,*]

[a] Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas, Ocampo y Esmeralda, S2000EZP Rosario, Argentina
[b] Universidad Nacional de Rosario, FCEyA, Pellegrini 250, S2000BTP Rosario, Argentina

ARTICLE INFO

Article history:
Available online 27 June 2014

Keywords:
Intelligent data analysis
Temporal abstraction
Temporal reasoning
Description Logic
Ontology
Semantic sensor web

ABSTRACT

In the past years, the large availability of sensed data highlighted the need of computer-aided systems that perform intelligent data analysis (IDA) over the obtained data streams. Temporal abstractions (TAs) are key to interpret the principle encoded within the data, but their usefulness depends on an efficient management of domain knowledge. In this article, an ontology-based framework for IDA is presented. It is based on a knowledge model composed by two existing ontologies (Semantic Sensor Network ontology (SSN), SWRL Temporal Ontology (SWRLTO)) and a new developed one: the Temporal Abstractions Ontology (TAO). SSN conceptualizes sensor measurements, thus enabling a full integration with semantic sensor web (SSW) technologies. SWRLTO provides temporal modeling and reasoning. TAO has been designed to capture the semantic of TAs. These ontologies have been aligned through DOLCE Ultra-Lite (DUL) upper ontology, boosting the integration with other domains. The resulting knowledge model has a modular design that facilitates the integration, exchange and reuse of its constitutive parts. The framework is sketched in a chemical plant case study. It is shown how complex temporal patterns that combine several variables and representation schemes can be used to infer process states and/or conditions.

1. Introduction

Sensing and communication technologies, such as "wireless sensor networks", have witnessed explosive growth in the recent past. These technologies are empowering information systems from many domains such as health care, industrial control systems and environmental monitoring, to collect and store large volumes of data (Molina & Flores, 2012). In addition, the recently developed sensor web technologies enable sensor measurements from all kind of sources to be available for sharing through web services (Sheth, Henson, & Sahoo, 2008; Ye et al., 2012). However, although these data is a very valuable asset for process analysis and supervision, it is usually not properly exploited. Therefore the need for computer-aided systems that extract useful knowledge from that large amount of available data becomes evident.

Intelligent data analysis (IDA) is and emergent research field that aims at feeling the gap between data generation and data comprehension, providing an efficient mean of matching raw data to process knowledge Lavrač, Keravnou, and Zupan (1997). IDA is directed toward application of knowledge for data interpretation. It takes advantage from different tools such as statistics, pattern recognition, data mining, machine learning and data abstraction; to discover the principles that are encoded within the sensed data.

The choice of a particular data representation has a large impact on efficiency and simplicity of IDA tasks (Lin, Keogh, Lonardi, & Chiu, 2003). Qualitative representation and reasoning has proven to be an excellent practice to embrace dynamic processes complexity since it relies on abstracted views of signals behavior, instead on just raw sensor outputs. These views are temporal abstractions (TAs) that interpret past and present states and trends that are relevant for the given set of goals. TAs are interval based representations having a wide range of complexity, from relatively simple level shifts to trend compound abstractions based on combinations of more primitive abstractions (Calbimonte, Yan, Jeung, Corcho, & Aberer, 2012; Cheung & Stephanopoulos, 1990; Janusz & Venkatasubramanian, 1991; Lin et al., 2003; Meléndez & Colomer, 2001; Molina & Flores, 2012; Shahar, 1994). One important advantage of using qualitative representation (QR) for IDA is that it enables artificial intelligence symbol-based reasoning, which brings a transparent way of capturing the process condition. TAs can be interpreted by matching against predefined templates

* Corresponding author at: Centro Internacional Franco Argentino de Ciencias de la Información y de Sistemas, Ocampo y Esmeralda, S2000EZP Rosario, Argentina.
E-mail addresses: roda@cifasis-conicet.gov.ar (F. Roda), musulin@cifasis-conicet.gov.ar (E. Musulin).

http://dx.doi.org/10.1016/j.eswa.2014.06.033
0957-4174/

or guidelines (Haimowitz & Kohane, 1996; Seyfang & Miksch, 2004), or reasoned within a higher context (Bellazzi, Larizza, Magni, Montani, & Stefanelli, 2000).

Very different domains such as medicine (Stacey & McGregor, 2007) and Process System Eng. (PSE), have shown an special interest in IDA solutions. Works from these areas present different but complementary approaches for the analysis and interpretation of sensor measurements. Qualitative Trend Analysis (QTA) is an outstanding method widely studied in PSE field (Cheung & Stephanopoulos, 1990; Gamero, Melendez, & Colomer, 2011; Janusz & Venkatasubramanian, 1991; Maurya, Paritosh, Rengaswamy, & Venkatasubramanian, 2010; Venkatasubramanian, Rengaswamy, Kavuri, & Yin, 2003; Villez, Venkatasubramanian, & Rengaswamy, 2013; Villez et al., 2013). QTA can identify a set of basic trends in the measured variable by looking at it derivatives signs. We call these kind of method *shape-based* since they depend on the shape of the observed signals. Note that these methods do not make use of domain knowledge in the abstraction process.

However, in medicine, IDA approaches rely more on heuristic knowledge (Esfandiary, Babavalian, Moghadam, & Tabar, 2014; Stacey & McGregor, 2007) that is captured by different representation schemes ranging from elaborated ontologies (Shahar, 1994; Shahar & Musen, 1996) to more simple point schemadata (Seyfang & Miksch, 2004; Seyfang et al., 2001). These techniques involve taking patient raw time-stamped data and using domain knowledge to generate temporal abstractions; such as "severe anemia for 3 weeks in the context of administering the drug AZT". Shahar (1994) presented one of the first works in that matter; the author calls it the "Knowledge-based Temporal abstraction theory (KBTA)". The core of KBTA is a set of five inference mechanisms (Temporal Context Formation, Contemporaneous Abstraction, Temporal Inference, Temporal Interpolation, Temporal Pattern Matching) supported by an ontology with 5 concepts (Primitive parameters, Events, Contexts, Abstract Parameters, and Patterns). KBTA was implemented in a computer program called RÉSUMÉ. It uses a frame-based languages to formalize the ontology concepts, and inference mechanisms are encoded with external rule-based tools. A shortcoming of this approach is that the knowledge specification and the inference mechanisms are decoupled. Furthermore, these formalisms are not sufficiently expressive to represent complex domain knowledge and temporal entities. In RÉSUMÉ, inferences are also bounded by constraints such as the *closed-word* and the *unique-name* assumptions that may lead to wrong deductions.

O'Connor, Hernandez, and Das (2011a) deal with these issues by means of semantic web technologies, which take advantage of Description Logic (DL) reasoning (Krötzsch, Simancik, & Horrocks, 2012). In particular, the authors make use of OWL[1] (Web Ontology Language) and SWRL[2] to integrate the specification and querying components of the KBTA methods. They showed how SWRL built-in functions are suitable to implement the five KBTA inference mechanisms, including a method for adding TAs on line. However, besides the OWL based implementation, the authors do not introduce changes to the original lightweight ontology of KBTA.

As shown, the above IDA approaches are developed to work with specific TA schemes and therefore they are not flexible enough to support different time series representations and abstraction levels. On the other hands, none of these efforts have considered an important goal of IDA: to extract knowledge from different data sources.

To address this issues, in this work an ontology-based framework to support intelligence data analysis of sensed data is presented. It relies in a novel knowledge model composed by four ontologies: Temporal Abstractions Ontology (TAO), Semantic Sensor Network ontology (SSN) (Compton et al., 2012), SWRL Temporal Ontology (SWRLTO) (O'Connor & Das, 2011) and DOLCE Ultra-Lite (DUL) (Masolo, Borgo, Gangemi, Guarino, & Oltramari, 2003). The framework uses temporal reasoning to search and classify qualitative temporal patterns, that help to infer the process state or condition.

The knowledge model is able to manage several TA schemes; both, knowledge-based temporal abstractions and shaped-based temporal abstractions are supported. This not only brings flexibility, but also enhances the analytical skills of the tool, as dynamic properties can be analyzed by combining different abstract views of the process.

Another important feature of the proposed framework is the integration with SSN that enables a full compatibility with sensor web technologies. Through it, the knowledge base have access to information about sensors and sensor observations from all kind of sources.

DOLCE Ultra-Lite has been employed as the upper-level ontology of the proposed framework. As a featured foundational ontology, DOLCE eases the understanding of the model and boosts future integrations with a large amounts of domains ontologies that are based on it.

Finally, we present a set of ontological correspondences for a full semantic alignment between the aforementioned ontologies.

The paper is organized as follows. Section 2 describes the main components, features and requirement of the proposed IDA framework. Section 3 introduces the ontologies used. In Section 4 these ontologies are semantically aligned to form a novel knowledge model. In Section 5, the framework is illustrated in a simple example in the PSE domain. Finally, in Section 6 concluding remarks are given.

2. Ontology-based framework for IDA

Any IDA system must cope with three interrelated issues: data validation, data representation and data interpretation. In this work, these issues were met by a novel IDA framework that takes advantage of the latest semantic technologies.

Fig. 1 depicts the work-flow in the proposed framework. The IDA process starts with the sensing tasks over the dynamic system under study. It is usually implemented by automatic sensor devices, but it can also be achieved manually. In each observation, a sensor measures a system property and provides an estimated value, a time stamp and some contextual data such as a measurement quality estimation. These observation records are validated and stored in large repositories (usually implemented as time series databases). Validation involves some data pre-processing tasks such as noise reduction, outlier detection and the rectification of input errors. To properly evaluate the stored measurements, the corresponding sensors metadata must be attached to observation records. This information should include sensors precision, operation range, Eng. units, etc. Data acquisition and validation tasks are high domain-dependent and therefore they must be specifically designed for each application environment. For this reason, no implementation guidelines are proposed here for these tasks. Instead, a reusable formal knowledge model for data representation is presented (see Section 3). It provides a foundational structure for the knowledge base (KB) where all the acquired knowledge is stored. As depicted in the figure, the conceptualization is composed by a set of three main ontologies:

1. An ontology of measurements, incorporated by semantically annotating the observation records.

[1] http://www.w3.org/TR/owl-features/.
[2] http://www.w3.org/Submission/SWRL/.

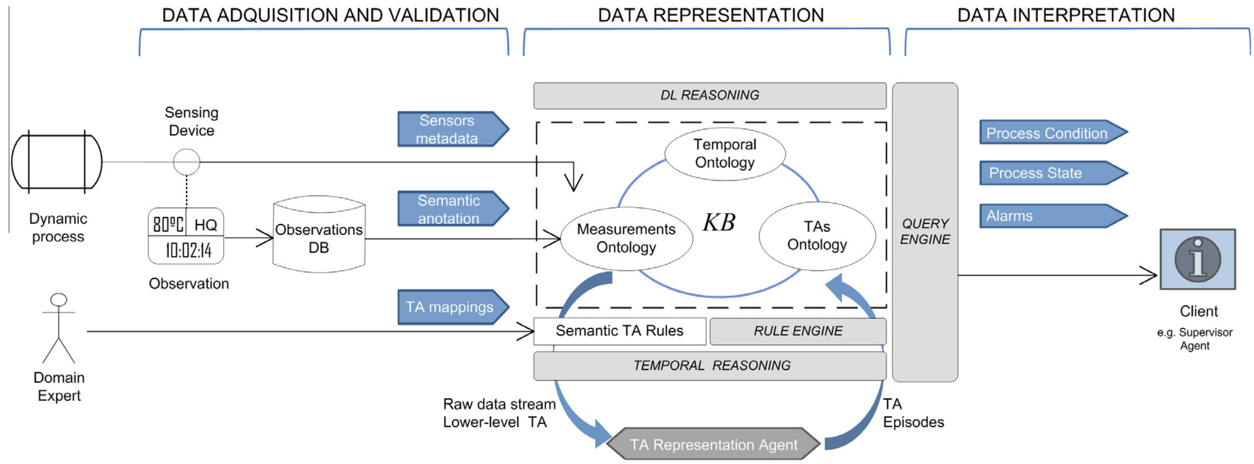

Fig. 1. Work-flow in the proposed IDA framework.

2. An ontology of TAs for different representation schemes and abstraction levels.
3. An ontology of temporal concepts that assists reasoning tasks.

Since data came from heterogeneous sources, the entities of the KB must be integrated in a consistence model expressed in a computer-interpretable language.

In this framework the responsibility of populate the KB with qualitative episodes lies in a specialized agent (i.e. *TA representation Agent*) that must be trained to detect and interpret process trends using a specific abstraction method. It takes raw data and/or TAs as inputs and generates qualitative episodes in a higher abstraction level. One important source of knowledge for TAs creation came from the domain experts, but it can also be obtained by machine learning methods. The expertise of these actors are valuable to state a mappings from the operational data to the abstract concepts. These expert knowledge can also be integrated in the KB as Horn-like rules that aid the TA representation tasks.

The reasoning tasks are performed by different inference engines. A DL reasoner maintains the KB consistency and provides a sound and complete classification scheme with satisfactory measures of computer complexity. Nevertheless, the DL inference routines are not able to process temporal dimensions on concepts or relations. Therefore, a temporal reasoning layer must be incorporated. Temporal reasoning is essential for IDA representation and interpretation tasks because it enables transparent and consistent temporal pattern matching. In addition, TA representation agents may need to implement additional reasoning routines in order to follow up specific representation methods.

Finally, a query engine is needed so as clients could perform queries about the measurements and qualitative representations of the process variables and states.

2.1. Implementation technology

The chart in Fig. 2 shows the central components and technologies proposed to implement the aforementioned tasks. As any knowledge based system, this IDA framework involves a knowledge base, an inference engine or reasoner and some interfaces to other computer systems or users.

Knowledge Base — It is composed by: (1) The classes, properties and constraints that define the domain concepts (i.e. TBox statements), (2) A set of ontological assertion axioms that describe the instances for particular applications (i.e. ABox statements), (3) A set of if-then rules to infer new facts. Ontological statements are implemented in OWL2, a sound and complete language with a formal semantic based in *SROIQ* Description Logic (DL) (Horrocks, Kutz, & Sattler, 2006). Rules expressed as OWL axioms are limited by some restrictions that enable the language to be decidable.[3] Thus, Semantic Web Rule Language (SWRL) (Horrocks et al., 2004) has been considered to formulate the KB rules.

Inference engine — Any off-the-shelf DL reasoner can be used to process OWL axioms; most of then are based on the Tableaux Decision algorithm (Horrocks & Sattler, 2007). However, SWRL expressions require a SWRL-enabled reasoner like Pellet or Kaon, or the addition of a rule engine such as Jess or Drool.

Interface — The Semantic Query Web Rule Language (SQWRL) (O'Connor & Das, 2009), a SQL-like language for querying ontologies, is proposed as the interface for knowledge extraction.

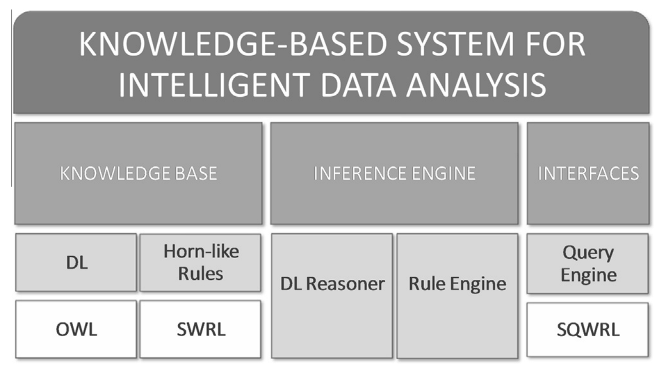

Fig. 2. Implementation technology for the proposed system.

3. Ontological modeling

Ontological modeling has been carried out following the five ontological principles stated by Gruber (1995): *Clarity* to

[3] http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/#Global_Restrictions_on_Axioms_in_OWL_2_DL.

communicate the intended meaning of defined terms, *Coherence* to sanction inferences that are consistent with definitions, *Extensibility* to anticipate the use of the shared vocabulary, *Minimal Encoding Bias* to be independent of the symbolic level and *Minimal Ontological Commitments* to make as few claims as possible about the world.

The development process has been based on the methodology presented by Noy and McGuinness (2001). According to this, the following tasks were accomplished:

Step 1. Determination of the scope and domain of the ontology.
Step 2. Search and evaluation of existing ontologies that fit in the scope.
Step 3. Construction a new conceptualization to consider concepts not covered by existing ontologies.
Step 4. Alignment of the selected ontologies in a single and consistent knowledge model.
Step 5. Model implementation using OWL and SWRL.

Regarding to the scope, three essential knowledge requirement have been identified to support IDA.

1. A scheme to capture the quantitative raw data produced by sensing tasks. In this regard, it is useful to consider the established standards to share measurements and sensor data.
2. A scheme for temporal modeling and reasoning. It must deal with the representation of temporal entities (e.g. time instants, periods, duration, etc.) and with the interpretation and processing of temporal relations (e.g. before, after, during, etc.).
3. A scheme to store and manage complex temporal abstractions. Whit such a scheme, the IDA system can store and manage qualitative representation of dynamic properties at different abstraction levels. This scheme should include information about the methods and actors responsible for the produced abstractions.

Several conceptualizations that partially met the above modeling requirement have been taken into account. Next subsections present a brief description of the reused and new conceptualization and explains how they were aligned to built a novel ontology.

3.1. Sensor measurements modeling

The input of an IDA system is a stream of time-stamped values usually produced by one or more sensors mounted to observe a significant process variable. Sensors are different to other technologies, such as services in service-oriented architectures, because of its event based nature and the temporal and spatial dimensions that need to be considered. Consequently, in recent years there has been a rising interest in ontologies and other semantic technologies to improve the integration and communication between sensor networks (Compton, Henson, Neuhaus, Lefort, & Sheth, 2009; Ye, Coyle, Dobson, & Nixon, 2007). The basic idea under these approaches is annotating sensor data with spatial, temporal, and thematic semantic metadata that increase interoperability as well as provide contextual information.

One of the most remarkable works in that matter is the Semantic Sensor Network (SSN) ontology (Compton et al., 2012). SSN (see Fig. 3) targets at the formal and machine-processable representation of sensor capabilities, properties, observations and measurement processes. SSN allows the network, its sensors and the resulting data to be organized, installed and managed, queried, understood and controlled through high-level specifications. The SSN ontology has implicated a large conceptualization effort to merge sensor-centric and observation-centric approaches. In order to do this, SSN leverages the Sensor Web Enablement (SWE) (Botts, Percivall, Reed, & Davidson, 2008) standard. It was formalized using OWL2 and is available as a single OWL File at http://purl.oclc.org/NET/ssnx/ssn.

The SSN ontology is resolved around the *Stimulus-Sensor-Observation pattern* (SSO) presented by Stasch, Janowicz, Bröring, Reis, and Kuhn (2009). This pattern was developed following the principle of minimal ontological commitments to make it reusable for a variety of application areas. It introduces a minimal set of classes and relations centered around the notions of *stimuli*, *sensor*, and *observation*. *Stimuli* are detectable changes in the environment that a sensor observes to infer information about environmental properties and construct features of interest. Thus, *Stimuli* play the role of a link to the physical environment. In SSN, *Stimuli* is considered as an *Event* and is represented by the equivalent classes *ssn:Stimuli* and *ssn:SensorInput*.

Observation is defined as "a *Situation* in which a *Sensing* method has been used to estimate or calculate a value of a *Property* of a *Feature Of Interest*". The class *ssn:Observation* provides the structure to represent a single observation, hence it is related to a single measurement (i.e. class *ssn:SensorOutput*) and attributed to a single property (i.e. classes *ssn:Property* and *ssn:FeatureOfInterest*) and to a particular *ssn:Sensor*.

A *ssn:FeatureOfInterest* represents a real world phenomena, it can be events (e.g. a reaction) or objects (e.g. an equipment) but not qualities. *ssn:Property* is defined as the quality of the phenomena to be measured (such as temperature or pressure).

The result of the sensing process is modeled by the class *ssn:SensorOutput*. Concrete data values (e.g. "30 °C", "60 mph", etc.) are represented through a *hasValue* relationships to an *ObservationValue* and then through the data property *DUL:hasRegionDataValue* to a XML Scheme data type.

Information about observation times is represented by means of two object properties, *ssn:observationSamplingTime* and/or *ssn:observationResultTime*, that link a *ssn:Observation* with a time instant. The first property points the time at which the observation has been made, while the second property points at the time at which the result is available. It is also possible to provide some information on the quality of an observation through the *ssn:qualityOfObservation* property.

According to the SSO pattern, a *ssn:Sensor* is defined as an entity that transforms an incoming stimulus into another representation. To do so, sensors implements a specific method that results in the estimation, or calculation of the value of the phenomena. This process is represented by the class *ssn:Sensing* and is related with the sensor by the relation *ssn:implementedBy* and its inverse *ssn:implements*. In most of the cases sensors are implemented with physical devices. In SSN these are represented by the class *ssn:SensingDevice* which is a subclass of both *ssn:Sensor* and *ssn:Device*. The *ssn:Device* class describes an abstract device and inherits all the properties of the class *ssn:System* (i.e. sub-components, platform to which a system is attached, deployment in which a system participates, operating and survival range). SSN also includes several classes that describe the measurement capabilities of the *ssn:Sensor* such as *ssn:Sensitivity*, *ssn:Accuracy* and *ssn:MeasurementRange*.

3.2. Temporal modeling and reasoning

A proper temporal model must be a principled model that enforces a consistent representation of temporal information in the system. Temporal modeling is key in IDA because both sensor measurements and qualitative representations need to be placed on a temporal dimension so that reasoners can interpret and process them.

Although, OWL is a powerful standardized technology for representing information and reasoning with it, this has very limited support for temporal information modeling. In fact, OWL only

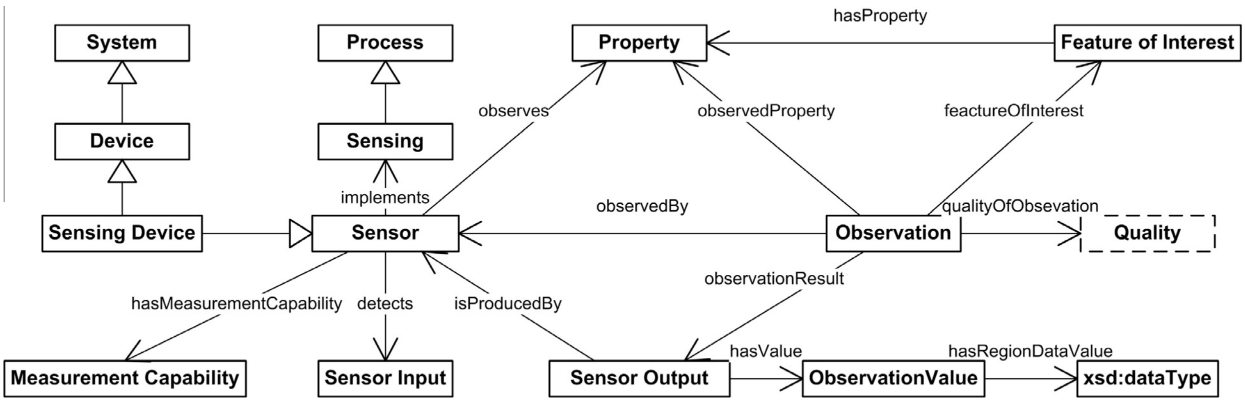

Fig. 3. Partial overview of the SSN conceptualization (UML diagram).

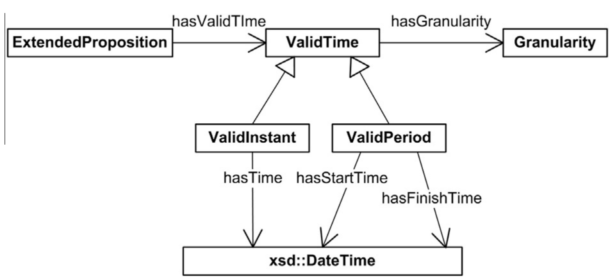

Fig. 4. SWRL Temporal Ontology.

Table 1
Allen's temporal relations. i and j are temporal intervals. i starts at the time point i_s and ends at i_e while j starts at j_s and ends at j_e. It is assumed that $i_s < i_e$ and $j_s < j_e$. Operators labeled with $(*)$ are meaningful to relate both instants and intervals.

Relation	Pictorical Example	Relations on Endpoints	Inverse
i before j (*)	i / j	$i_e < j_s$	*j after i (*)*
i meets j	i / j	$i_e = j_s$	*j metBy i*
i overlaps j	i / j	$i_s < j_s < i_e \wedge i_e < j_e$	*j overlappedBy i*
i starts j (*)	i / j	$i_s = j_s \wedge i_e < j_e$	*j startedBy i (*)*
i during j (*)	i / j	$i_s > j_s \wedge i_e < j_e$	*j contains i (*)*
i finishes j (*)	i / j	$i_s < j_s \wedge i_e = j_e$	*j finishesBy i (*)*
i equals j	i / j	$i_s = j_s \wedge i_e = j_e$	

allows data values to be typed as basic XML Schema dates, times or durations.[4] Additionally, SWRL includes some basic operators for manipulating time points, but time intervals are no supported. As a consequence, several novel approaches have been presented to overcome these OWL shortcomings (Artale & Franconi, 2000; Anagnostopoulos, Batsakis, & Petrakis, 2013; Batsakis & Petrakis, 2010; Batsakis, Stravoskoufos, & Petrakis, 2011; O'Connor & Das, 2011; Papadakis, Stravoskoufos, Baratis, Petrakis, & Plexousakis, 2011).

In this work, the *SWRL Temporal Ontology* (SWRLTO) (O'Connor & Das, 2011) has been used because it is a lightweight solution that provides a simple scheme to operate over temporal information in queries and rules. SWRLTO is an open source OWL ontology that can be layered on existing ontologies without requiring they to be significantly rewritten. In fact, it has been successfully applied to several works on medicine (O'Connor et al., 2011a, 2011b; Subirats et al., 2013; Weichert, Mertens, Walczak, Kern-Isberner, & Wagner, 2013).

SWRLTO is based on the *valid-time* temporal model (Snodgrass, 1995). In this model, every temporal fact can be associated with an instant or an interval denoting the *Fact's Valid-Time*. These temporal references are known as the *Valid-Time* as the fact is held to be true or valid during that period. No conclusions can be made about the fact for time regions outside of this. The SWRL Temporal Ontology provides OWL with the tools for representing all the entities defined by this model. Fig. 4 shows its mains classes and properties using UML notation. The class *swrlto:ValidTime* has two subclasses: *swrlto:ValidInstant* and *swrlto:ValidPeriod*. A *Valid Instant* denotes a point on a time-line. A *Valid Period* models the time between two instants. These are specified by the *swrlto:hasStartTime* and *swrlto:hasFinishTime* date-time properties. The *swrlto:Granularity* class represent the unit of measure for temporal datum.

[4] XML Schema. http://www.w3.org/TR/xmlschema11-1/, 2009.

Regarding the temporal reasoning scheme, SWRLTO implements a set of SWRL built-in predicates to handle temporal relations. Most of them are based on the Allen algebra (Allen, 1983), which defines 13 temporal operations that allow to relative positioning any pair of intervals by comparing it starting and ending points (see Table 1). Although this algebra was not originally intended to relate an interval with an instant or even two instants, SWRLTO built-in predicates enable it for some operators. Additionally, SWRLTO provides some built-in functions to perform granularity conversion and duration calculations at varying granularity.

3.3. TA modeling

Literature does not present many works about ontology-based TA modeling. The conceptualization proposed by Shahar and Musen (1996) for the KBTA method is perhaps the most relevant contribution among the few works on that matter. Here, a novel ontology to formally represent the semantic of temporal abstractions is presented. Thought it is founded on the basic principles of KBTA's inference mechanisms, the proposed approach aims at a more flexible modeling that also supports shaped-based methods (such as QTA or QRT) as well as ad hoc machine learning-based methods. The presented, is a DL formal model that leverages the reasoning capabilities of semantic web technologies and that it is easily integrable to others OWL ontologies. It is called *Temporal Abstractions Ontology* (TAO).

Fig. 5 shows a diagram of the proposed conceptualization using UML notation. Temporal abstractions are qualitative representations (QRs) that provide the means to integrate domain context

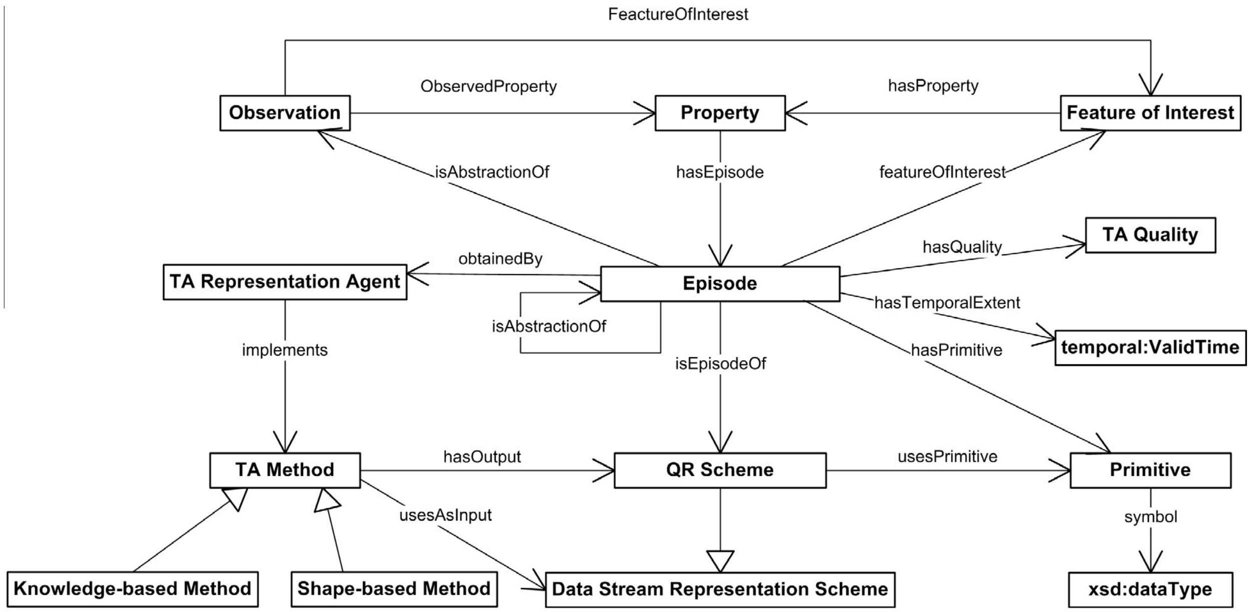

Fig. 5. Temporal Abstraction Ontology.

to analyze real time data. QRs are defined as the description of a time series by means of contiguous *Episodes* (Villez et al., 2013). In TAO, QR Schemes are modeled by the class *tao:QR Scheme* which is linked to a set of primitives forming the used alphabet, and to the method/s that can be employed to obtain it.

Episodes are temporal abstractions of data stream slices obtained by a heuristic or formal method. They are formally defined as a set of two elements: a time interval, named temporal extent, and a qualitative context, providing the temporal extension with significance (Williams, 1986).

In the ontology, this concept is modeled by the class *tao:Episode*. The temporal extent is given by a *swrlto:ValidTime* while the qualitative context is given by a *tao:Primitive*. Like *ssn:Observation*, *tao:Episode* is associated with the *property* of the *tao:feature of interest* that is abstracted by the episode.

Primitives are the elemental symbols of the alphabet used for a given QR scheme. In the ontology, these are specified by the relation *tao:usesPrimitive* from *tao:QR Scheme* to *tao:Primitive*. For instance, in many derivatives-based representation primitives are described by alphabetic characters such as A, B, C, etc. In OWL, these symbols can be stored as XML scheme data linked with the primitive through the relation *tao:Symbol*. By definition, an episode of a QR is characterized by a single primitive. The primitive of an episode is stated by the object property *tao:hasPrimitive*.

The class *tao:TA Representation Agent* has been defined in order to trace the abstraction process. A TA representation agent is an entity representing the author of a given sequence of episodes. It is not limited to software agents and it can also be a human expert performing a heuristic analysis. Anyhow, every TA agent performs a specific sequence of tasks that result in a qualitative representation of the input data stream. This process is modeled by the class *tao:TA Method*.

The input of a TA method is a data stream representation that can be in a quantitative form, such as the actual sensor measurements, or in a qualitative form, such as the episodes in a lower abstraction level (see the relation *tao:usesAsInput* from *tao:TA Method* to *tao:Data Stream Representation Scheme*). For that reason, the range of the object property *tao:isAbstractionOf* is defined as a disjunction of *tao:Observations* and *tao:Episodes*. The latter let to build qualitative episodes from a set of episodes in a lower abstraction level, forming a multilayer abstraction hierarchy (see the recursive relation on the class *tao:Episode*).

The output of the TA method must be a qualitative representation. In fact, the relation *tao:hasOutput* (and its inverse) allows to state which method/s can be implemented in order to obtain a particular QR Scheme. Two categories of TA methods have been defined by subsuming the class *tao:TA Method* with the classes *tao:Shape-based Method* and *tao:Knowledge-based Method*.

Shape-based methods depend on the shape of the observed signals. They can identify a finite set of shapes in the observed variable (e.g. a rising or a decreasing trend), once detected a symbol is associated with the correspondent time interval. Almost no domain knowledge or expertise is required to obtain these abstractions. In fact, they can be applied to different domains without any modification in its routine. Example of shape-based methods are QTA (Cheung & Stephanopoulos, 1990; Janusz & Venkatasubramanian, 1991; Meléndez & Colomer, 2001) and the slope-based representation (Calbimonte et al., 2012).

Knowledge-based methods must be specifically defined for each domain. The abstractions are real interpretations of the process dynamic that rely on the expertise captured by the representation agent. They are strongly dependent on the context and need a sound analysis of historical data. These methods are usually implemented with a set of *If-Then* rules or may be supported by model-based tools. Examples of Knowledge-based TA methods are can be found in Shahar and Musen (1996), Montani, Leonardi, Bottrighi, Portinale, and Terenziani (2013) and Musen, Middleton, and Greenes (2014).

Quality is an indicator of the data reliability highly dependent on the method followed to obtain it. Many modern sensing systems provide a measure of the observation quality[5] based on the sensor measurement properties regarding the observed value. Observation quality is usually expressed by categories like *good*, *bad* and *uncertain*, sometimes followed by a code giving some insight (e.g. *Not Connected*, *Configuration Error*, *EU Units Exceeded*, etc.). Like observations, qualitative episodes are estimations of a property

[5] Observation Quality is considered by the O&M standard through the attribute *resultQuality*, equivalent to *ssn:qualityOfObservation*.

under study represented in a particular abstraction level, thus they should also have an associated quality value. In fact, *TA quality* should be tied to the percentage of good data available to their calculus. In the proposed framework, it can be stated by a set of SWRL rules so that inference engines can automatically deduce them. In that way, the quality of the data sources is dragged into the abstraction process.

In the proposed framework, the arguments to set the quality of TA Episodes are expressed by a set of SWRL rules, thus enabling inference engines to deduce them.

It is important to note that with a proper instantiation, the proposed conceptualization enables different representation approaches such as the Triangular Episodes of Cheung and Stephanopoulos (1990) or Symbolic Aggregate Approximations of Lin et al. (2003). These representations can be stored in the same KB as alternative views of the process under study.

Once the ontologies that support the knowledge structure have been determined, they must be integrated into a single conceptualization. This procedure is explained in the following section.

4. Semantic alignment

The proposed framework reuses the aforementioned knowledge models. However, since these ontologies have been developed to meet specific needs under particular domains and following different approaches, they had to be semantically aligned. Ontology alignment can be defined as a set of correspondences between two ontologies (Shvaiko & Euzenat, 2005). A correspondence states a relation hold between two entities, one from each ontology, such as equivalence ($=$); more general ($\sqsupseteq$) or disjointness ($\perp$).

In this work, a semantic alignment method based on an upper-level ontology has been employed. Since upper-level ontologies (i.e. foundational ontologies) are logic formalization of very general concepts shared across different domains, in ontology alignment, they act as external sources of common knowledge. As a semantic technique, it maps concepts based on the analysis of interpretations, and not on labels, as in syntactic matching. In particular, SSN, TAO and SWRLTO have been aligned using DOLCE + DnS Ultralite ver. 3.27 (DUL), a simplified version of DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) (Masolo et al., 2003). It has been chosen for the following reasons:

- It is a comprehensive ontology that covers many domains and fields of research.
- It has a rigorous conceptualization that facilitates the interpretation of the domain primitives.
- It boss further alignment and reuse as many ontology-based frameworks are already based on it (e.g. WordNet).
- SSN, one of the biggest component of this framework is already aligned with DUL.

The alignment has been performed as follows: first, the upper entities of DUL and SWRLTO have been aligned in order to provide DOLCE with a time model. Then, SSN and TA have been aligned with the resulting upper-level ontology.

4.1. SWRLTO-DUL alignment

In this subsection, the semantic alignment between SWRLTO and DUL is given using DL notation. Since DUL is founded upon the concepts of the DOLCE Lite-Plus, the possible mappings between SWRLTO and DOLCE have been also analyzed. However, it must be stressed that only the SWRLTO-DUL mappings are needed for implementing the proposed framework. alignment of classes,

$$swrlto:ValidTime \equiv dul/dolce:Time\ Interval \tag{1}$$
$$swrlto:Granularity \sqsubseteq dul:Unit\ of\ measure \tag{2}$$
$$swrlto:Granularity \sqsubseteq dolce:non\text{-}physical\text{-}object \tag{3}$$
$$swrlto:ExtendedProposition \sqsupseteq (dul:Quality \sqcup dul:Situation) \tag{4}$$
$$swrlto:ExtendedProposition \equiv dolce:Perdurant \tag{5}$$

alignment of object properties,

$$swrlto:hasValidTime \sqsupseteq dolce:temporal\text{-}location \tag{6}$$
$$swrlto:hasValidTime \sqsupseteq dul:is\ observable\ at \tag{7}$$
$$swrlto:hasGranularity \sqsubseteq dul/dolce:has\ region \tag{8}$$

alignment of data properties,

$$swrlto:hasStartTime \sqsubseteq dul:has\ interval\ date \tag{9}$$
$$swrlto:hasFinishTime \sqsubseteq dul:has\ interval\ date \tag{10}$$
$$swrlto:hasTime \sqsubseteq dul:has\ interval\ date \tag{11}$$

It must be noted that this alignment is not straightforward nor the correspondences are unique as it has to deal with two important issues: (1) SWRLTO and DUL concepts are in a similar abstraction level (i.e. they contain foundational concepts) and the formal semantic of SWRLTO is not provided. (2) These ontologies consider different representation of temporal information. DOLCE (and thus DUL) adopts a *4D-fluent* approach (Masolo et al., 2003) while SWRLTO is presented as a *reification solution* (O'Connor & Das, 2011).

In 4D-fluent, concepts in time are represented as 4-dimensional objects with the 4th dimension being the time. It divides the world in two basic categories: *endurants* and *perdurants* (also called fluent or occurrent). The former represent information that do not change over time while the latter are entities that extend in time by accumulating different "temporal parts". In this way, changes occur on the properties of the temporal part keeping the entities of the static part unchanged (Hawley, 2001).

On the other hand, the knowledge model of SWRLTO is quite flexible and it does not have constraints that prevent from the use of other similar approaches. In fact, the meaning of a *dolce: Perdurant* is consistent with the definition of the concept *swrlto: Extended Proposition* "entities that can extend over time". For this reason, these concepts have been mapped as equivalent.

Another important correspondence has been defined between the classes *swrlto:ValidTime* and *dul:Time Interval*. Even thought *swrlto:ValidTime* seems to be a more general concept, as it involves time instants and periods, a *Time Interval* in DOLCE is defined as "any region in a dimensional space that aims at representing time". Due to this semantic agreement, a correspondence has been stated between them.

Finally, it is stressed that thanks to this alignment any domain ontology, new or existent, based on DOLCE can incorporate temporal reasoning capabilities. Furthermore, since the alignment works on upper-level concepts, it can be achieved without any modification of the domain ontologies. This is possible because, every *dolce:perdurant* member having a *dolce:Time interval* as *temporal location* can now be interpreted as an *swrlto:Extended Proposition* having a *swrlto:Valid Time*, thus enabling SWRLTO built-in functions to be used in rules or queries to reason over temporal facts.

4.2. SSN-DUL alignment

The W3C incubator group developed SSN subsuming its concepts and properties to DUL, so these ontologies are already aligned. The following is a list of the most important correspondences for classes that were presented in Compton et al. (2012). Here, they are transcribed using DL notation.

$$ssn : Sensor \sqsubseteq dul : PhysicalObject \quad (12)$$
$$ssn : Observation \sqsubseteq dul : Situation \quad (13)$$
$$ssn : SensorOutput \sqsubseteq dul : InformationObject \quad (14)$$
$$ssn : SensorInput \sqsubseteq dul : Event \quad (15)$$
$$ssn : Process \sqsubseteq dul : Method \quad (16)$$
$$ssn : FeatureOfInterest \sqsubseteq (dul : Event \sqcup dul : Object) \quad (17)$$
$$ssn : Property \sqsubseteq dul : Quality \quad (18)$$
$$ssn : ObservationValue \sqsubseteq dul : Region \quad (19)$$
$$ssn : Device \sqsubseteq dul : DesignedArtifact \quad (20)$$

4.3. TAO-DUL alignment

The following correspondences have been defined to align TA and DUL. This has been carried out consistently with the SSN-DUL alignment.

alignment of classes,

$$tao : TA\ Method \sqsubseteq dul : Method \quad (21)$$
$$tao : Episode \sqsubseteq dul : Situation \quad (22)$$
$$tao : Primitive \sqsubseteq dul : Pattern \quad (23)$$
$$tao : TA\ Representation\ Agent \sqsubseteq dul : Agent \quad (24)$$
$$tao : TA\ Quality \sqsubseteq dul : Social\ attribute \quad (25)$$
$$tao : Data\ Stream\ Representation\ Scheme \sqsubseteq tao : QR\ Scheme \sqsubseteq dul : Theory \quad (26)$$

alignment of object properties,

$$tao : obtainedBy \sqsubseteq dul : is\ conceptualized\ by \quad (27)$$
$$tao : hasEpisode \sqsubseteq dul : is\ object\ included\ in \quad (28)$$
$$tao : hasPrimitive \sqsubseteq dul : satisfies \quad (29)$$
$$tao : usesPrimitive \sqsubseteq dul : has\ component \quad (30)$$
$$tao : isAbstractionOf \sqsubseteq dul : has\ constituent \quad (31)$$
$$tao : isEpisodeOf \sqsubseteq dul : is\ described\ by \quad (32)$$
$$tao : hasQuality \sqsubseteq dul : has\ region \quad (33)$$
$$tao : implements \sqsubseteq dul : is\ described\ by \quad (34)$$
$$tao : usesAsInput \sqsubseteq dul : is\ related\ to\ description \quad (35)$$
$$tao : hasOutput \sqsubseteq dul : is\ related\ to\ description \quad (36)$$

The main guidelines of this alignment are grounded in the "*Description and Situation*" design pattern included in DUL.[6].

dul:Situation is defined as "A view, consistent with ('satisfying') a Description, on a set of entities. It can also be seen as a 'relational context' created by an observer on the basis of a 'frame' (i.e. a Description)." In TAO this is precisely the role played by a *tao:Episode* (see Eq. (22)). In fact, *tao:Episode* is a situation detected in a dynamic property expressed by means of an abstract concept (i.e. the *tao:Primitive*). That is, an episode has presence meanwhile the process dynamic *satisfies* the description enclosed in the associated primitive. Here, the *TA Representation Agent* is the observer who checks the presence of qualitative episodes.

tao:Primitive is subsumed by *dul:Pattern* (see Eq. (23)) because it closely fits the *dul:Pattern* definition. According to DUL, a pattern is "any invariance detected from a data set, or from observation; also, any invariance proposed based on top–down considerations." The definition adds: "an occurrence of a pattern is an 'observable', or detected Situation". In the same way, an occurrence of a *tao:Primitive* is a detected *tao:Episode*. To complete the alignment, and considering that a *tao:Episode* is a *dul:Situation*, it must satisfies the following DUL existential restriction,

$$Situation \sqsubseteq \exists satisfies.Description \quad (37)$$

this can be proved using Eqs. (23) and (29), and taken into account that *dul:Pattern* is a subclass of *dul:Description*,

$$Pattern \sqsubseteq Relation \sqsubseteq Description. \quad (38)$$

The mapping presented in Eq. (26) gives to *tao:QR Scheme* a formal context to state its semantic. In DUL, a *dul:Theory* is a *dul:Description* that represents a set of assumptions for describing something. These assumptions are the "components" of the theory which must be instances of the class *dul:Relation*. This means that *dul:Theory* members have to satisfy the restriction,

$$Theory \sqsubseteq \exists has\ component.Relation \quad (39)$$

As it was enunciated in Section 3.3, a *tao:QR scheme* encloses a set of primitives that provide a qualitative context to interpret the dynamic of a process. In other words, primitives are the component of the theory enclosed in a QR Scheme,

$$QR\ Scheme \sqsubseteq \exists use\ primitive.Primitive \quad (40)$$

Furthermore, note that *tao:Primitive* is a subclass of *dul:Relation* (see Eqs. (23) and (38)) and *tao:use primitive* is a subproperty of *dul:has component* (see Eq. (30)). Therefore Eq. (39) is true for every *tao:QR Scheme* class member.

In Eq. (25), *tao:TA Quality* is considered a *dul:Social attribute* because *tao:TA Quality* is an attribute of a qualitative episode and, in the terms of DUL, *tao:Episode* is essentially a *Social Object* (i.e. $tao : Episode \sqsubseteq dul : Situation \sqsubseteq dul : Social\ Object$).

By Eq. (27), it can be enunciated that "*an Episode is conceptualized by an Agent*". In DUL, *dul:is conceptualized by* is a relation stating that an *Agent* is internally representing a *Description*. e.g., 'John believes in the conspiracy theory'; 'Jacques assumes all swans are white'. Similar formulations can be made about the building of qualitative episodes. For instance, "a software agent believes that the plant reactor is faulty", "the clinician thinks the patient has a cold".

In Eq. (31), *tao:isAbstractionOf* is subsumed by *dul:has constituent*. *"Constituency"* is a DUL relation that depends on some layering of the world described by the ontology (e.g. the layering social–mental–biological–physical). A constituent is a part belonging to a lower layer (e.g. the persons constituting a social system). In the TA domain, layers are built on the basis of a given set of abstraction goals. As it was explained in Section 3.3, qualitative episodes *are constituted by* entities of a lower abstraction level (Observation or other Episodes). *tao:isAbstarctionOf* is the TAO relation that allows to link these episodes across different abstraction layers, thus enabling the building of multilayer abstraction hierarchies.

One important implication of this alignment is that *TAO* gets from DOLCE the *4D-fluent* (perdurantist) approach for representing temporal information. *tao:Episode* is the main *perdurant* entity on TAO; it becomes a perdurant because *dul:Situation* is essentially perdurant (see Eq. (22)). On the other hand, classes such as *tao:Primitive*, *ssn:Property* and *tao:TA Agent* play an endurant role because they do not change over time.

4.3.1. TAO-SSN alignment

The semantic alignment with DUL presented in the previous sections is an essential prerequisite to consistently integrate TAO, SSN and SWRLTO into a single KB. However, some extra correspondences have been added to improve the association between TAO and SSN.

In order to minimize coupling between KB modules and to increase the whole KB cohesion, some SSN entities have been

[6] http://ontologydesignpatterns.org/wiki/Submissions:DescriptionAndSituation

copied into TAO instead of just to import them. In particular, *ssn:Observation*, *ssn:Property*, *ssn:Feature of Interest* and its main objects properties have been replicated in TAO (see Fig. 5). As a result, two occurrences of each ones appear when integrating the ontologies. In order to state that these entities refer to the same things the following equivalence mappings need to be added.

$$tao : Observation \equiv ssn : Observation \tag{41}$$

$$tao : Property \equiv ssn : Property \tag{42}$$

$$tao : Feature\ of\ Interest \equiv ssn : Feature\ of\ Interest \tag{43}$$

$$tao : feature\ of\ interest \equiv ssn : feature\ of\ interest \tag{44}$$

$$tao : has\ property \equiv ssn : has\ property \tag{45}$$

$$tao : observedproperty \equiv ssn : observed\ property \tag{46}$$

It must be noted that these duplications acts at the conceptual level (class level) and they not involve data redundancy (i.e. instances are not duplicated).

4.3.2. Refining the temporal model alignment

In Section 4.2, the alignment between SWRLTO and DUL has been presented. In this section, the alignments DUL-SSN and DUL-TAO are completed to ensure that the valid-time temporal model from SWRLTO is available for SSN and TAO. Entities such as *tao:Episode* or *ssn:Observation* must be linked with *valid-times* representing their temporal location.

SSN defines only two object properties to temporarily locate observations: *ssn:observation result time* and *ssn:observation sampling time*. However, SSN does not include a model for representing time, in fact the range of these properties is not provided. When aligning SSN and DUL, the authors stated,

$$ssn : observation\ sampling\ time \sqsubseteq dul : has\ region \tag{47}$$

$$ssn : observation\ result\ time \sqsubseteq dul : has\ region \tag{48}$$

dul:has region is defined in DUL as a general relation that links entities (*class dul:Entity*) with data regions (*class dul:Region*).

In order to facilitate the integration of SSN and SWRLTO through DUL, and also to improve the SSN-DUL alignment, in this work the following correspondences are proposed,

$$ssn : observation\ sampling\ time \sqsubseteq dul : is\ observable\ at \tag{49}$$

$$ssn : observation\ result\ time \sqsubseteq dul : is\ observable\ at \tag{50}$$

These axioms are added to the aligned ontologies without altering or contradicting the original SSN-DUL alignment. This is true because Eqs. (49) and (50) are consistent with Eqs. (47) and (48) and due to the fact that,

$$dul : is\ observable\ at \sqsubseteq dul : has\ region \tag{51}$$

In addition, a similar mapping is defined between TAO and DUL:

$$tao : hasTemporalExtent \sqsubseteq dul : is\ observable\ at \tag{52}$$

Tanks to the inheritance relationships between the aligned ontologies, SSN and TAO can leverage the temporal reasoning capabilities of SWRLTO. Fig. 6 shows how temporal properties are inferred by the reasoner. *tao:Episode* and *ssn:Observation* inherit the temporal role and the properties of *swrlto:Extended Proposition* through *dul:Situation* (see the mapping Eqs. (4), (13) and (22)).

Due to the mapping Eqs. (7), (49), (50) and (52), the properties *ssn:observation sampling time*, *ssn:observation result time* and *tao:hasTemporalExtent* became subproperties of *swrlto:hasValidTime*. In that way, the property *swrlto:hasValidTime* can be employed to link the instances of *tao:Episode* or *ssn:Observation* with time instants (*swrlto:ValidInstant*) or periods (*swrlto:ValidPeriod*).

As a consequence, the SWRLTO built-in predicates that allow to reason over temporal facts (i.e. instances of *swrlto:ExtendedProposition*) became available to operate over episodes and observations.

5. Illustrative example

In order to illustrate the framework capabilities to support IDA tasks, it has been used to supervise a Continuously Stirred-Tank Reactor (CSTR). CSTR is a common reactor widely used in industrial chemical plants. In the example, it is assumed that a set of sensors are mounted to measure some important variables that allows to control the process (e.g. tank temperature, effluent concentration, tank level, etc). The implementation required the construction of an application ontology and its instantiation with domain knowledge.

In this example, the ontology editor Protégé 3.5[7] has been used with the SWRL tab plug-in activated, which enables the processing of SWRL expressions using the Drools rule engine. HermiT has been used for DL reasoning. In the following section it is explained how measurements are stored and managed in the knowledge base for this case study.

5.1. KB instantiation

Consider the observations realized by a temperature sensor (T1) located in the reactor tank. T1 is sampled each minute, Fig. 7 shows an interpolated view of three samples of T1 together with its qualitative representation (i.e. the primitive "A"). In the KB, this information is represented as a set of interconnected instances, which are distributed through the four aligned ontologies.

Observations *T_O1*, *T_O2* and *T_O3* correspond to SensorOutputs *T_SO1*, *T_SO2* and *T_SO3* respectively. Measured values are represented by instances of the class *TemperatureValue* (a *ssn:ObservationValue* subclass). *T_episode001* represents the signal slice that has been abstracted as the "A" primitive. This episode has been identified and instantiated by an external software agent (*QRPT_Agent*) that implements the "Qualitative Representation of Process Trends" (*QRPT_Method*) due to Cheung and Stephanopoulos (1990). Since in QRPT the primitives are identified with the sign of the first and second derivatives, *A_primitive* is asociated with XML literals that store these signs and the symbol ("A"). Note that this primitive must be linked by the property *usesPrimitive* with the method used to extract the episode (*QRPT_Method*). Otherwise the reasoning results in a logic inconsistency.

The relationship *isAbstractionOf* can be deduced by the DL reasoner using the time reference and the involved property, or it may be provided by the external agent. In the latter case, the DL reasoner check their consistency. In addition, both Observations and its qualitative representation are associated to the same *Property* (*ReactorTemperature*) and the same *Feature of Interest* (*Reactor*) to be consistent.

All temporal references are represented by instances of SWRLTO. *T_episode001* spans over the time interval (*vp1*) starting at 09:00 and finishing at 09:03. Likewise, each *Observation* (*T_O1*, *T_O2* and *T_O3*) is associated with the time instants of sensing (*t1*, *t2* and *t3*) by the property *observationSampleTime*. The association labeled «*contains*», between *vp1* and the *Valid Instants* represents the Allen temporal relation which are inferred by the rule engine through the SWRLTO built-in predicates. These are implicit facts stating that *t1*, *t2* and *t3* are temporally inside the interval *vp1*.

[7] http://protege.stanford.edu/.

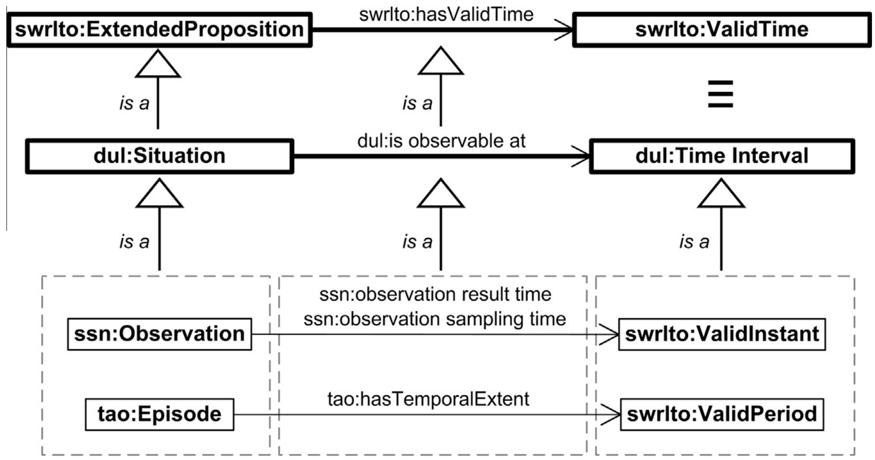

Fig. 6. The scheme shows the inheritance relationships that make the SWRLTO *valid-time* model available for both SSN and TAO ontologies.

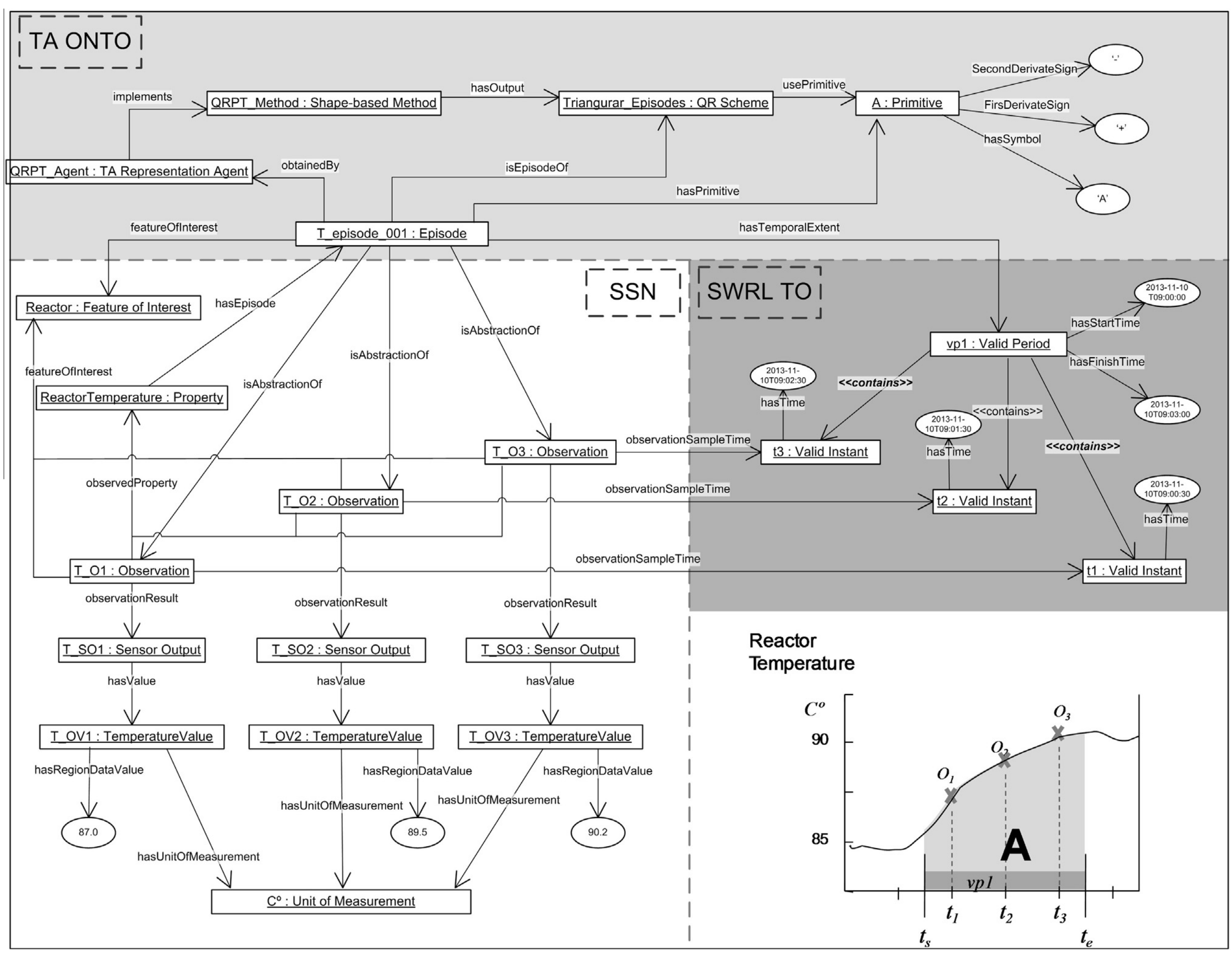

Fig. 7. Ontology instantiation to represent a qualitative episode in the Reactor Temperature.

5.2. Knowledge exploitation

Temporal reasoning joined to TA management provides a valuable tool for data analysis as the expressive power of rules and queries is improved. The following example show the use of rules to check temporal consistency. As explained above, a data stream abstracted by an episode must be temporally located inside the time period of the episode, otherwise data is inconsistent. The next rules let to verify this constraint.

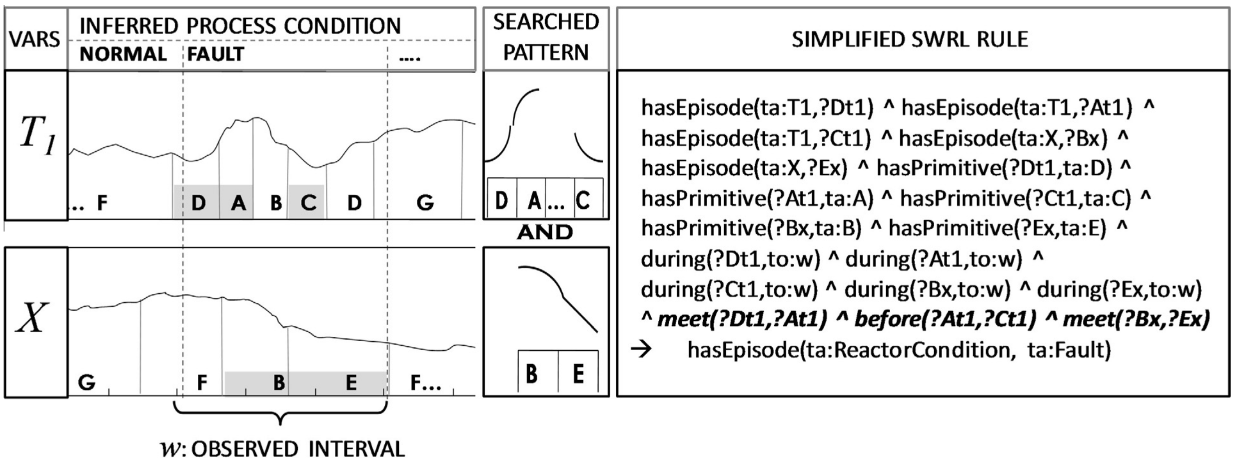

Fig. 8. Multivariate qualitative temporal reasoning example.

```
isAbstractionOf (?a,?o)^ hasValidTime (?a,?ta)^ has
ValidTime (?o,?to)^ before (?ta,?to) -> isNotAb-
stractionOf (?a,?o).
isAbstractionOf(?a,?o)^ hasValidTime(?a,?ta)^ has
ValidTime(?o,?to)^ after(?ta,?to) -> isNotAbstract
ionOf(?a,?o).
```

These rules state that, if an entity (a sensor measurement or a qualitative episode) (`?o`) is temporally located after or before the Valid Period of its abstraction (`?a`), then the fact `isNotAbstractionOf` is inferred between the correspondent instances. Since an axiom in the ontology states that `isAbstractionOf` is disjoint with `isNotAbstractionOf`, the DL reasoner can detect the contradiction.

Temporal operators are also helpful in queries and rules to support multivariate data analysis. The presented approach let temporal patterns (e.g. a fault pattern) to be defined and searched using high-level expressions that are closer to process experts.

Consider for example that we are interested in analyzing the dynamic of T1 in periods of high concentration in the effluent F (i.e. High X). Its must be noted that this query involves two properties and two QR schemes. The dynamic of T1 may be studied through its QRPT episodes. However, concentrations are better interpreted as deviation classes such as High, Low, etc. The upper and lower bounds of each class can be set by a Hazard identification analysis or by a PAA-based method such as SAX (Lin et al., 2003). Then, the concentration episodes can be achieved by a simple limit checking algorithm.

```
hasEpisode (tao:effluentConcentration,?ce)^ has
Primitive (?ce,tao:x_Hight)^
hasEpisode (tao:reactorTemperature,?te)^ obtained
By (?te, tao:qrtp_agent)^ during (?te,?ce)^
hasPrimitive (?te,?tp)^ hasStartTime (?te,?teS)^
hasFinishTime (?te, teF?)^ hasSymbol (?tp,?s)
->select (?s,?teS,?teF)
```

It returns the symbols (?*s*) and the temporal locations (?*teS* and ?*teF*) of the QRPT episodes of T1.

Multivariate temporal reasoning is also valuable to infer a process condition. Fig. 8 shows an example of a rule expression that states a Fault Condition based on the observed dynamic.

Here, a particular combination of episodes must be found on T1 and in the concentration (X) in the same time interval w. As it is shown in the T1 search pattern, a flexible formulation is allowed, since any episode could be placed between *A* and *C*.

6. Conclusions

In this paper an ontology-based framework to support intelligent data analysis (IDA) of sensed data is presented. It takes advantages of semantic technologies to arrive at high-level qualitative descriptions about the state or condition of a dynamic process of interest.

This function is crucial to analyze and to interpret raw sensor data in several domains; such as in detecting and diagnosing faults in industrial plants, in medical patients monitoring and in warning severe weather conditions.

The main contribution of this work is a novel knowledge model that integrates four featured ontologies: TAO, SSN, SWRLTO and DUL. This includes the development of a new Temporal Abstraction Ontology (TAO).

Unlike other IDA approaches, this framework has been designed to support both knowledge-based (e.g. KBTA) and shaped-based (e.g. QTA, QRPT) temporal abstractions together. This brings flexibility to data representation thus enhancing analytical possibilities.

This work leverages the SSN initiative to integrate semantic data about sensors and sensor observations of all type through the web. The SSN alignment assures a full compatibility with the open standards of OCG for sensor web and enables abstraction tasks to be traced, as qualitative episodes remain linked to the sensor outputs and the TA methods.

The proposed tool is able to monitor dynamic processes through time by mean of a lightweight solution for temporal modeling and reasoning (i.e. SWRLTO). An important consequence of handling temporal relations (e.g. before, during, overlap, etc) is that qualitative temporal pattern can be formulated like regular expression using incomplete sequences of symbols. Temporal Patterns can be placed in both rules and queries, and are interpreted by any off-the-shelf SWRL-enabled reasoner.

The domain ontologies (TAO, SSN, SWRLTO) have been aligned through the foundational concepts of DOLCE Ultra-Lite (DUL). Since DOLCE is a rigorous conceptualization widely used in several domains, its alignment facilitates the interpretation of the domain primitives and boosts future integrations with other ontologies. Additionally, the KB has been developed following a modular design that minimizes the coupling between the ontologies. As a consequence, any component of the model can be reused individually. For instance, another DUL-based application may take advantages of the DUL-SWRLTO alignment for incorporating temporal modeling and reasoning.

The presented example illustrated how this approach can be applied for supervising a chemical process. It has been shown that it is able to maintain temporal consistency among entities from different abstraction levels. But more importantly, it has been shown how a process condition (e.g. a fault in an industrial plant) can be inferred by tracking multivariate qualitative temporal patterns.

Instead of the promising results, some weaknesses of the framework can be identified. For future works we are considering the following issues. First, we plan to incorporate pattern recognition methods based on similarity measures. These methods can enhance queries by relaxing the matching functions using the semantic distances between the qualitative episodes.

With regards to temporal reasoning we are considering to improve the used approach with *stream reasoning* capabilities. Stream reasoning is a subject of topical interest for the Semantic Web that aims at providing high-level skills for processing time stamped data. For instance, with this technology the IDA system could query about the number of *A* episodes in the last hour. Another hot spot of future work will be the extension of the IDA framework to support multiresolution data analysis (e.g. wavelet transform, multigrid method).

Acknowledgments

The authors thank the financial support of CONICET (National Scientific and Technical Research Council), UNR-FCEIA (Rosario National University), and ANPCYT (Agencia Nacional de Promoción Científica y Técnica) from Argentina.

References

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM, 26*(11), 832–843.

Anagnostopoulos, E., Batsakis, S., & Petrakis, E. G. (2013). Chronos: A reasoning engine for qualitative temporal information in owl. *Procedia Computer Science, 22*, 70–77.

Artale, A., & Franconi, E. (2000). A survey of temporal extensions of description logics. *Annals of Mathematics and Artificial Intelligence, 30*(1–4), 171–210.

Batsakis, S., & Petrakis, E. G. (2010). Sowl: Spatio-temporal representation, reasoning and querying over the semantic web. In *Proceedings of the sixth international conference on semantic systems* (pp. 15). ACM.

Batsakis, S., Stravoskoufos, K., & Petrakis, E. G. (2011). Temporal reasoning for supporting temporal queries in owl 2.0. In *Knowledge-based and intelligent information and engineering systems* (pp. 558–567). Springer.

Bellazzi, R., Larizza, C., Magni, P., Montani, S., & Stefanelli, M. (2000). Intelligent analysis of clinical time series: An application in the diabetes mellitus domain. *Artificial Intelligence in Medicine, 20*(1), 37–57.

Botts, M., Percivall, G., Reed, C., & Davidson, J. (2008). Ogc® sensor web enablement: Overview and high level architecture. In *GeoSensor networks* (pp. 175–190). Springer.

Calbimonte, J., Yan, Z., Jeung, H., Corcho, O., & Aberer, K. (2012). Deriving semantic sensor metadata from raw measurements. In *Proceedings of the fifth international workshop on semantic sensor networks.* URL: <http://ceur-ws.org/Vol-904/paper4.pdf>.

Cheung, J., & Stephanopoulos, G. (1990). Representation of process trends – Part I. A formal representation framework. *Computers & Chemical Engineering, 14*(4/5), 495–510.

Compton, M., Henson, C. A., Neuhaus, H., Lefort, L., & Sheth, A. P. (2009). A survey of the semantic specification of sensors. In *Second international workshop on semantic sensor networks, CEUR-WS* (pp. 17–32).

Compton, M., Barnaghi, P., Bermudez, L., García-Castro, R., Corcho, O., & Cox, S., et al. (2012). The ssn ontology of the w3c semantic sensor network incubator group. In *Web semantics: Science, services and agents on the world wide web* (Vol. 17, pp. 25–32).

Esfandiary, N., Babavalian, M. R., Moghadam, A.-M. E., & Tabar, V. K. (2014). Knowledge discovery in medicine: Current issue and future trend. *Expert Systems with Applications, 41*(9), 4434–4463.

Gamero, F. I., Melendez, J., & Colomer, J. (2011). Qssi: A new similarity index for qualitative time series. Application to classify voltage sags. *Applied Artificial Intelligence, 25*(2), 141–162.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing. *International Journal of Human–Computer Studies, 43*(5), 907–928.

Haimowitz, I. J., & Kohane, I. S. (1996). Managing temporal worlds for medical trend diagnosis. *Artificial Intelligence in Medicine, 8*(3), 299–321.

Hawley, K. (2001). *How things persist*. Oxford University Press.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M., et al. (2004). Swrl: A semantic web rule language combining owl and ruleml. In *W3C Member submission* (Vol. 21, p. 79).

Horrocks, I., Kutz, O., & Sattler, U. (2006). The even more irresistible SROIQ. In *Proc. of the 10th int. conf. on principles of knowledge representation and reasoning (KR2006)* (pp. 57–67). AAAI Press.

Horrocks, I., & Sattler, U. (2007). A tableau decision procedure for $\mathcal{SHOIQ}$. *Journal of Automated Reasoning, 39*(3), 249–276.

Janusz, M. E., & Venkatasubramanian, V. (1991). Automatic generation of qualitative descriptions of process trends for fault detection and diagnosis. *Engineering Applications of Artificial Intelligence, 4*(5), 329–339.

Krötzsch, M., Simancik, F., & Horrocks, I. (2012). A description logic primer. CoRR abs/1201.4089.

Lavrač, N., Keravnou, E. T., & Zupan, B. (1997). *Intelligent data analysis in medicine and pharmacology: An overview*. Springer.

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. (2003). A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the eighth ACM SIGMOD workshop on research issues in data mining and knowledge discovery* (pp. 2–11). ACM.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., & Oltramari, A. (2003). Wonderweb deliverable d18, ontology library (final). ICT project.

Maurya, M. R., Paritosh, P. K., Rengaswamy, R., & Venkatasubramanian, V. (2010). A framework for on-line trend extraction and fault diagnosis. *Engineering Applications of Artificial Intelligence, 23*(6), 950–960.

Meléndez, J., & Colomer, J. (2001). Episodes representation for supervision. Application to diagnosis of a level control system. In *Workshop on principles of diagnosis DX* (Vol. 1, pp. 119–126).

Molina, M., & Flores, V. (2012). Generating multimedia presentations that summarize the behavior of dynamic systems using a model-based approach. *Expert Systems with Applications, 39*(3), 2759–2770.

Montani, S., Leonardi, G., Bottrighi, A., Portinale, L., & Terenziani, P. (2013). Supporting flexible, efficient, and user-interpretable retrieval of similar time series. *IEEE Transactions on Knowledge and Data Engineering, 25*(3), 677–689.

Musen, M. A., Middleton, B., & Greenes, R. A. (2014). Clinical decision-support systems. In *Biomedical informatics* (pp. 643–674). Springer.

Noy, N. F., & McGuinness, D. L., et al. (2001). Ontology development 101: A guide to creating your first ontology.

O'Connor, M. J., & Das, A. K., 2009. Sqwrl: A query language for owl. In *Proc. of sixth OWL: Experiences and directions workshop (OWLED2009).*

O'Connor, M. J., & Das, A. K. (2011). A method for representing and querying temporal information in owl. In *Biomedical engineering systems and technologies* (pp. 97–110). Springer.

O'Connor, M. J., Richards, A., Martins, S., Bingen, M., Tu, S. W., & Das, A. (2011). Sweetinfo: a web-based system for visualizing and querying temporal data. In *Fifth international RuleML2011@BRF challenge* (Vol. 799, pp. 63–70). CEUR-WS, Fort Lauderdale, USA.

O'Connor, M. J., Hernandez, G., & Das, A. (2011a). A rule-based method for specifying and querying temporal abstractions. In *Artificial intelligence in medicine* (pp. 255–259). Springer.

Papadakis, N., Stravoskoufos, K., Baratis, E., Petrakis, E. G., & Plexousakis, D. (2011). Proton: A prolog reasoner for temporal ontologies in owl. *Expert Systems with Applications, 38*(12), 14660–14667.

Seyfang, A., & Miksch, S. (2004). Advanced temporal data abstraction for guideline execution. *Studies in Health Technology and Informatics*, 88–102.

Seyfang, A., Miksch, S., Horn, W., Urschitz, M. S., Popow, C., & Poets, C. F. (2001). Using time-oriented data abstraction methods to optimize oxygen supply for neonates. In *Artificial intelligence in medicine* (pp. 217–226). Springer.

Shahar, Y. (1994). A knowledge-based method for temporal abstraction of clinical data (Ph.D. thesis). School of Medicine, Stanford University.

Shahar, Y., & Musen, M. A. (1996). Knowledge-based temporal abstraction in clinical domains. *Artificial Intelligence in Medicine, 8*(3), 267–298.

Sheth, A., Henson, C., & Sahoo, S. S. (2008). Semantic sensor web. *IEEE Internet Computing, 12*(4), 78–83.

Shvaiko, P., & Euzenat, J. (2005). A survey of schema-based matching approaches. In *Journal on data semantics IV* (pp. 146–171). Springer.

Snodgrass, R. T. (1995). *The TSQL2 temporal query language* (Vol. 330). Springer.

Stacey, M., & McGregor, C. (2007). Temporal abstraction in intelligent clinical data analysis: A survey. *Artificial Intelligence in Medicine, 39*(1), 1–24.

Stasch, C., Janowicz, K., Bröring, A., Reis, I., & Kuhn, W. (2009). A stimulus-centric algebraic approach to sensors and observations. In *GeoSensor networks* (pp. 169–179). Springer.

Subirats, L., Ceccaroni, L., Gómez-Pérez, C., Caballero, R., Lopez-Blazquez, R., & Miralles, F. (2013). On semantic, rule-based reasoning in the management of functional rehabilitation processes. In *Management intelligent systems* (pp. 51–58). Springer.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S. N., & Yin, K. (2003). A review of process fault detection and diagnosis. Part III: Process history based methods. *Computers & Chemical Engineering, 27*, 327–346.

Villez, K., Rosén, C., Anctil, F., Duchesne, C., Vanrolleghem, P. a., & Jan. (2013). Qualitative representation of trends (QRT): Extended method for identification of consecutive inflection points. *Computers & Chemical Engineering, 48*, 187–199.

Villez, K., Venkatasubramanian, V., & Rengaswamy, R. (2013). Generalized shape constrained spline fitting for qualitative analysis of trends. *Computers & Chemical Engineering, 58*, 116–134.

Weichert, F., Mertens, C., Walczak, L., Kern-Isberner, G., & Wagner, M. (2013). A novel approach for connecting temporal-ontologies with blood flow simulations. *Journal of Biomedical Informatics, 46*(3), 470–479.

Williams, B.C. (1986). Doing time: Putting qualitative reasoning on firmer ground. In *AAAI* (pp. 105–112).

Ye, J., Coyle, L., Dobson, S., & Nixon, P. (2007). Ontology-based models in pervasive computing systems. *The Knowledge Engineering Review, 22*(4), 315–347.

Ye, J., Dasiopoulou, S., Stevenson, G., Meditskos, G., Efstathiou, V., & Kompatsiaris, I. (2012). Semantically-enriched pervasive sensor-driven systems. *Semantic Web Interoperability, Usability, Applicability, 1*, 1–5<http://www.semantic-web-journal.net/sites/default/files/swj298.pdf>.