How to track absolutely everything

Werner CEUSTERS[1], Shahid MANZOOR
Ontology Research Group, New York State Center of Excellence in Bioinformatics and Life Sciences, University at Buffalo, USA

Abstract: The analysis of events prior to and during September 11 revealed that a smooth execution of the intelligence process is hampered by inadequate information sharing. This caused a rethinking of the intelligence process and a transition towards a 'Globally Networked and Integrated Intelligence Enterprise' with the goal that more detailed, tagged, and, therefore, traceable, information will reach those who need it, when they need it, and in a form that they can easily absorb. We present the referent tracking paradigm and its implementation in networks of referent tracking systems as an enabling technology to make this vision come true. Referent tracking uses a system of singular and globally unique identifiers to track not only entities and events in first-order reality, but also the data and information elements that are created to describe such entities and events in information systems. By doing so, it meets the requirements of the Nation's *Information Sharing Strategy*.

Keywords: referent tracking

1. Introduction

Intelligence, as defined by the Central Intelligence Agency (CIA), is '*the information our nation's leaders need to keep our country safe*' [1]. This information is produced by the US Intelligence Community (IC), i.e. the departments and agencies cooperating to fulfil the goals of Executive Order 12333 which stipulates that '*The United States intelligence effort shall provide the President and the National Security Council with the necessary information on which to base decisions concerning the conduct and development of foreign, defense and economic policy, and the protection of United States national interests from foreign security threats*' [2]. This is achieved through the performance of what is called the '*intelligence process*' which consists of five steps: (1) the determination of the information requirements, (2) the collection of raw data, (3) the processing of the raw data into forms that are more usable for intelligence analysts or other consumers, (4) the integration, evaluation and analysis of the data in order to generate reports satisfying the requirements, and (5) the dissemination of the results to the appropriate level [3]. This last step, typically, leads to new information requirements which initiate a new cycle of the intelligence process.

[1] Corresponding Author: Werner Ceusters, Ontology Research Group, New York State Center of Excellence in Bioinformatics and Life Sciences, 701 Ellicott street, Buffalo NY, 14203, USA; E-mail: ceusters@buffalo.edu

1.1. Challenges and barriers

Ideally, the information that is finally disseminated is (1) *reliable*, thus corresponding faithfully to what is the case in reality, (2) *complete*, such that nothing what is essential or required for the consumer to make adequate decisions is missing, (3) *relevant*, such that decisions can be made efficiently, and (4) *timely*, guaranteeing that decisions can be made early enough for the resulting actions to have the desirable effect. Unfortunately, this ideal is very hard to achieve because of many barriers and challenges [4]. A large number of these challenges are brought about by the multiplicity of agencies, organizational levels within these agencies and information consumers that are involved.

Although each step in the intelligence process comes with its own challenges, the multiplicity of involved actors affects primarily the information requirements assessment and the data-integration and analysis steps. So do the information requirements that a specific organizational level has to take into account not only consist of the external requirements put forward by the consumers to whom intelligence reports of a specific nature and content need to be delivered, but also of the internal requirements which determine what sorts of detailed information elements are required and accessible to provide high quality reports. The integration, evaluation and analysis step can be hampered by insufficient lower-level data (both quantitatively and qualitatively), wrong information, and lack of meaningful data linkage. The net effect is that the reliability, completeness and relevancy of the resulting conclusions suffer considerably.

Although these three notions are intuitively straightforward, they can be defined in various ways and for each such way, objective quantification is hard, if possible at all. Furthermore, these notions are not entirely independent from each other. Reliability, for instance, relates to *accuracy* which itself relates to relevancy: the more a measurement is accurate, the more reliable it seems to be, yet, the relevancy of it might diminish depending on the objectives of the intelligence effort: whereas providing information on the duration of intercontinental flights in minutes to compare the performance of foreign carriers with that of national ones seems reliable, accurate and relevant, doing so in hours is hardly reliable, while in seconds for sure not relevant. *Redundancy* of information elements within a collection of information will not harm the completeness and relevancy of that collection as a whole, but for sure the relevancy of the redundant elements themselves. At the other hand, from a second order perspective, the presence of redundant information, if obtained from various independent sources, might be an indication for the reliability of the collection.

1.2. Intelligence and Security Informatics

The analysis of events prior to and during September 11 revealed that a smooth execution of the intelligence process is hampered by inadequate information sharing [5]. Not only are there legal and cultural barriers to information sharing – the '*need-to-know*' culture during the Cold War is now recognized to be a handicap in dealing with terrorism and other asymmetric threats [6] – it is also technically very difficult to integrate and combine data that are stored in different database systems running on different hardware platforms and operating systems [7]. Although the Office of Homeland Security, in 2002, identified information sharing across jurisdictional boundaries of intelligence and security agencies as one of the key foundations for ensuring national security [8], the appropriate infrastructure is not yet there. This recognition led to the development of a new science: '*Intelligence and Security Informatics*' (ISI) [9], which is commonly defined as '*the study of the use and development of advanced information technologies, systems, algorithms, and databases for national- and homeland-security-related applications through an integrated technological, organizational, and policy-based approach*' [10].

ISI tries to overcome the barrier that data which reside in distinct data sources are organized in different schemas, and therefore are difficult to integrate. But still, once some sort of integration has been achieved, it remains often very hard to determine, for instance, whether two distinct pieces of information are about the same entity or which piece of information is correct when several pieces about the same entity can't be true at the same time. As an example, a case study in a local police department revealed that more than half of the suspects had either a deceptive or an erroneous counterpart existing in the police system: 42% of the suspects had records alike due to various types of unintentional errors, while about 30% had used intentionally a false identity [11]. Deception is in the context of ISI a very hard problem indeed; it is not limited to providing false identities, but includes also '*cognitive hacking*' which involves disinformation attacks on the mind of the end user of a networked computer system such as a computer connected to the Internet [12]. Identifying such attacks is crucial in an era in which the Intelligence Community seeks to make better use of Open Source Information (OSINT) [13].

1.3. Vision 2015

To further advance the modernization of the information technology within the Intelligence Community, the Office of the Director of National Intelligence [14] published in February 2008 its '*Information Sharing Strategy*' report [6], followed in July by the '*Vision 2015*' document [15]. They key idea, first introduced in the National Intelligence Strategy [16], is the move towards a '*Globally Networked and Integrated Intelligence Enterprise*' with the goal that more detailed, tagged, and, therefore, traceable, information will reach those who need it, when they need it, and in a form that they can easily absorb. Efforts in these directions are expected to create the ability to develop, digest, and manipulate vast and disparate data streams '*about the world as it is today*' by means of tags that enable the use of tools that can '*trace related data across our holdings, to mine the data, to test hypotheses and to suggest correlations*' in addition to '*measuring performance*' [15].

The key characteristics of the new information sharing model are [6]:

- C1. ***'responsibility to provide'***: sharing intelligence data while still addressing the need to protect privacy, civil liberties, and sources and methods;
- C2. ***enterprise-centric***: providing services across agencies, partners, and international borders for multiple mission use;
- C3. ***mission-centric***: able to adapt rapidly to changing needs and new partners;
- C4. ***information-centric***: security built into the data and environment using tags;
- C5. ***attribute-based***: access based on attributes that go beyond security classification (e.g. environmental, affiliation, mission focus, etc.);
- C6. ***data 'stewardship'*** (rather than data 'ownership'), focusing on quality and reusability of data rather than, but not excluding, protection.

1.4. Tagging, indeed, but what and how?

Because '*tagging*' seems to be an important part of the proposed solution to make this vision come true, the issues that we address here are (1) where the tags should come from, (2) what it is that should be tagged, and (3) according to what sort of logical schema data and tags should be organized in order for the data to track faithfully what is going on in the world. We argue, in response to each of the issues just mentioned, (1) that the tags should correspond to the terms (or codes) which are used as representations for universals and defined classes in realism-based ontologies, thus covering what is generic, (2) that what is tagged should not only be the data about first-order entities (persons, vehicle movements, parcels, disease outbreaks, …), but also how and by whom (and what) these data are generated and manipulated, and (3) that the data should be organized in a structure which mimics the structure of that part of reality that is described by the data and that is capable to reflect all sorts of changes that reality undergoes in the course of history.

2. Naive tagging

Today, information is primarily maintained in information systems which consist of data repositories that contain data in either unstructured form (such as free text or digital multi-media objects) or structured form, the latter being such that numerical information is expressed by means of numbers, and non-numerical information by means of codes or terms associated with what is commonly called '*concepts*', taken from different sorts of terminologies (such as vocabularies, nomenclatures, concept systems, and so forth) as they are offered in terminology servers. Since data in structured form are better suited to provide software agents with a deep understanding of what the data represent, considerable efforts are spent to turn unstructured data into structured data, at least partially. However, whether data are captured in structured form when entered, or rendered as such afterwards using text and image analytics software which add tags corresponding to concepts, current information systems exhibit at least two major shortcomings as far as concept-based tagging is concerned: (1) formal impreciseness about *what* is tagged, and (2) incompatibility of distinct tagging systems.

2.1. Missing the point(ers)

Mainstream information systems do not offer a mechanism to unambiguously determine in each individual case what entity in reality a concept from a terminology server is used to relate to. As a consequence, information systems thus conceived work with instances of data, but algorithms working on such data have no clue what the data are about, i.e. about what specific entity in reality each specific data-element contains the information.

If, for example, a driving license number is used in an information system, it is often not formally clear whether the number is used to denote the driving license of a person or that person itself.

As a further example, if in an information system the gender of a person is stated to be '*unknown*', then it is often not formally clear whether this means either (1) that the person does have a gender which is one of the scientifically known gender types such as female, male, mosaic, etc., but that information of the precise gender of that person is not available in that information system, or (2) that the gender of that person is known to be of a type which scientifically has not yet been determined. Another example is that if at a certain time the gender of a specific person is registered in some information system as '*male*', and at a later time as '*female*', then there is, under existing data storage paradigms, no way to derive from this change whether the change in the information system reflects (1) a change in reality, for instance, because the person underwent transgender surgery, (2) a change in what became known about reality: the person's gender might because of a congenital disorder not have been determinable at the time of birth, but only later after several investigations, or (3) that there was no change in reality or what we know about it, but that at the time of the first entry a simple mistake was made. One can even imagine a fourth possibility, namely that the meaning of the word '*female*' would have been changed. The latter might seem to be too far fetched – in fact, this did never happen for the words '*male*' and '*female*' – but there are several examples in the past that come close. The title '*Chief Executive Officer*', for instance, was introduced in Europe in the late eighties, replacing titles such as '*Director General*' or '*Managing Director*'. A change in title, in those days, for sure did not entail a change in position or power of the person to whom the new title was attributed.

These types of issues are insufficiently addressed in modern Semantic Web applications because they are not yet generally recognized: attempts to address them are sparse.

2.2. Missing semantics

The most recent hype in information system networking is semantic interoperability. By '*semantic interoperability*', it is meant the ability of two or more computer systems to exchange information and have the meaning of that information automatically interpreted by the receiving system accurately enough to produce useful results, as defined by the end users of both systems. Current attempts to achieve semantic interoperability rely on agreements about the meaning of so-called concepts stored in terminology-systems, such as nomenclatures, vocabularies, thesauri, or ontologies, the idea being that if all computer systems use the same terminology, they can understand each other perfectly. The reality is, however, that, rather than one such terminology being generally adopted, the number of terminology-systems with mutually incompatible definitions or non-resolvable overlap amongst concepts grows exponentially, thereby contributing more to the problem of semantic non-interoperability than solving it. Of course, ontologies developed for different purposes can only reasonably be expected to have partial overlap, but more efforts should be conducted to exploit overlap when resolvable.

3. Fundamentals of realism-based ontologies and data repositories

In contrast to traditional terminology approaches, the *realist* orientation in terminology and ontology is based on the view that terms in terminologies are to be aligned not on concepts but rather on entities in reality [17]. Central to this view are three assumptions [18]. The first is that reality exists objectively in itself, i.e. independent of the perceptions or beliefs of cognitive beings. Thus not only do a wide variety of entities exist in reality (human beings, terrorists, guns, attacks, countries, ...), but also how these entities relate to each other (that human beings are citizens of countries, that in most attacks guns are used, and so forth) is not a matter of agreements made by scientists or database modellers but rather of objective fact.

The second assumption is that reality, including its structure, is accessible to us and can be discovered: it is scientific research that allows human beings to find out what entities exist and what relationships obtain between them. It is intelligence analysis that allows analysts to find out which specific human beings are terrorists.

The third assumption is that an important aspect of the quality of an ontology or terminology is determined by the degree to which the structure according to which the terms are organized mimics the pre-existing structure of reality.

In the context of information systems, it means that an important aspect of the quality of an information system is determined by the degree to which (1) its individual representational units correspond to entities in reality, and (2) the structure according to which these units are organized mimics the corresponding structure of reality.

3.1. Faithful representations

The above assumptions form the basis for distinguishing between three levels of reality which have a role to play wherever ontologies are used as artifacts for annotation and tagging, and wherever automated or semi-automated reasoning is required to be able to deal with an overload of information, parts of which can be expected to be wrong. Ontologies and data repositories for the intelligence community are no exception to this.

The three levels are [18]:

- Level 1: the (first-order) reality 'in the field': the persons that are tracked, the events that are monitored, the users of the information system, and so forth;
- Level 2: the beliefs and cognitive representations of this reality embodied in observations and interpretations on the part of observers, data collectors, analysts and others;
- Level 3: the publicly accessible concretizations of such cognitive representations in representational artifacts of various sorts, of which ontologies, terminologies and data repositories are examples. Ontologies contain typically representations for what is *generic*, thus representing entities such as person, weapon, war, and so forth. Repositories cover what is *specific*, thus holding representations for entities such as President George W. Bush Jr., the gun that killed John F. Kennedy, The Gulf War, etc.

In line with the theory of granular partitions [19] we argue that complex representations should be composed in modular fashion of sub-representations built out of representational units that are assumed to correspond to portions of reality (POR). Some characteristics of the units in a representation created for intelligence purposes are:

- each such unit is assumed by the authors of the representation to be veridical, i.e. to conform to some relevant POR as conceived on the best understanding (which may, of course, rest on errors). Thus if in a data repository a representational unit standing proxy for a specific person is associated with the name 'George Bush', then, under the realist paradigm, we assume that a person with this name exists or has existed (that on the basis of the name only it cannot be determined which specific person is meant, does not make the unit non-veridical);

- several units may correspond to the same POR by presenting different though still veridical views or perspectives, for instance at different levels of granularity (one thing may be described both as being brown and as reflecting light of a certain wavelength, or one event as an event of administering and of consuming drugs);
- what units are included in a representation depends on the purposes which the representation is designed to serve.

3.2. Keeping track of changes

The real world is subject to constant change, and so also is our knowledge thereof. To keep track of these two sets of changes, any representation concerning a relationship between entities should be associated with at least the following pieces of information: (P1) an index for the time period during which the relationship obtains, (P2) an index for the time at which the representation is made, i.e. the time at which the relationship is (believed to be) known, (P3) an index for the time that piece of information is made available in the system, and (P4) an identifier standing proxy for the author of the representation.

Keeping track of these various types of information makes it possible not only to track reality faithfully from an individual analyst or agency perspective, but also to preserve the knowledge about what was known by whom and at what time after information which was residing originally in distinct systems becomes merged. It also allows to assess whether information is disclosed in a timely fashion.

Suppose, for instance, that at time t10 it is known by analyst A1 that suspect S was since t9 member of group G of possible terrorists, but that an entry to that effect in the information system of his agency is made available not earlier than at t11. Thus between t10 and t11, that information was not accessible. Furthermore, in reality, it might be that S was already member of G at t5. That information might have been known in another agency since t6, and made available at that time in their information system. When the information in the two systems becomes merged, for instance after the Vision 2015 situation becomes reality, it can still be assessed what was known at each point in time in each agency.

4. Fundamentals of Referent Tracking

Referent Tracking (RT) is a paradigm for information management that is distinct from other approaches in that each data element has to point to a portion of reality in a number of predefined ways (Figure 1). It has been introduced in the context of Electronic Health Record keeping [20], but its applicability is wider than that, examples being digital rights management [21] and corporate memories [22].

By '*portion of reality*' is meant any individual entity or configuration of entities standing in some relation to each other. By '*entity'* is meant anything that exists or has existed in the past, whatever its nature. A '*configuration'* is a portion of reality which is not an entity in its own right. Whereas a specific person, his or her activities, the social network he belongs to, the analyst examining information about that person, and that examination itself are each individual entities, the configuration that the activities of this person are being monitored by an intelligence agency, or his or her being part of that social network, is not. Another example of a configuration is the being of an engine in a car. Both that car and that engine are entities, but the fact that that engine is in that car, is not. If that engine would not be in the car, but, for instance be placed by a mechanic outside the car for repair purposes, still the very same entities (the car and the engine) would be involved, but there would be another configuration.

Within the RT paradigm, configurations are referred to by means of a data type called a '*RT-tuple*', whereas entities are represented by means of a data type called '*representation*'. Both data types come in several forms depending on the nature of the portion of reality they carry information about (see section 6).

RT, through its data types, allows also for the drawing of an explicit distinction made in Basic Formal Ontology (BFO) [23] between specific entities called '*particulars'* from generic entities called '*universals'*. Particulars are specific and unique entities, unique in the sense that they each occupy specific regions of space and time, and that nothing other than a specific particular can be that particular. Examples are concrete persons such as George W. Bush Jr. and George W. Bush's heart. Some particulars, such as each of four tanks in a specific squadron, may exactly look the same, but they are still distinct particulars. One can be destroyed, while the other three remain intact. For particulars of specific interest, such as persons, ships, and hurricanes, proper names are used to mark the importance of their individual identity. For other particulars, such as cars or pieces of complex equipment, serial numbers are used for unique identification purposes.

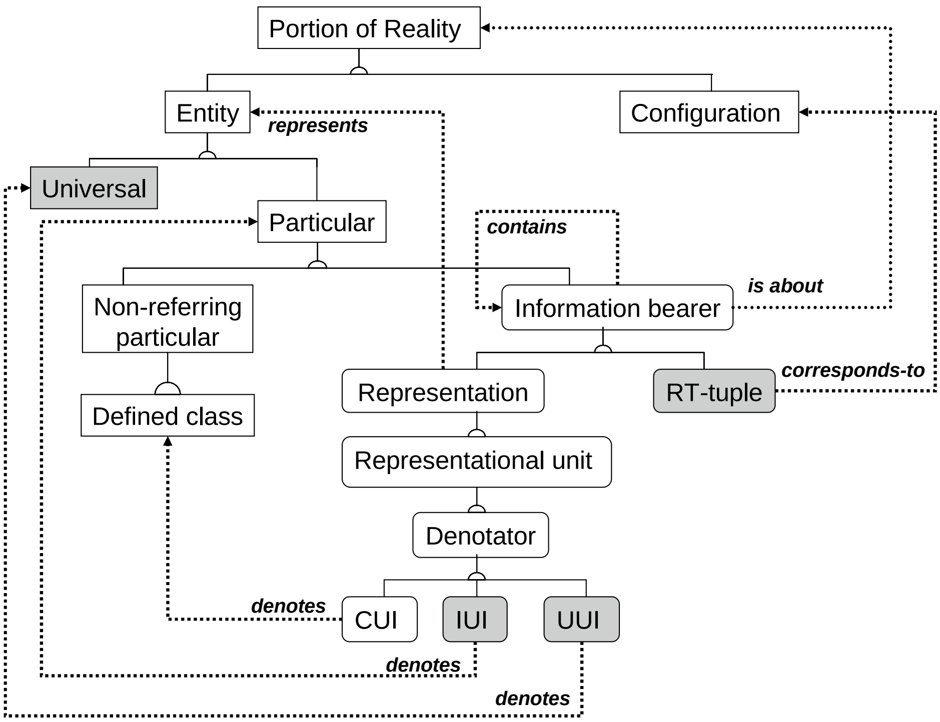

Figure 1: Reality and representations

Universals, in contrast, are such that they are (1) generic and (2) expressed in language by means of general terms such as '*person*', '*ship*', and '*car*', and (3) represent structures or characteristics in reality which are exemplified in an open-ended collection of particulars in arbitrarily disconnected regions of space and time.

Through yet other data types, RT makes explicitly the distinction between two sorts of particulars: those that are '*information bearers*', and those that are not; the latter called '*non-referring particulars*'. Whereas non-referring particulars belong exclusively to the first level of reality – they are pure first-order entities – information bearers play a role in both levels 1 and 3.

Examples of information bearers are a piece of paper containing a text about a person's educational background, and a digital object, such as an image of a person in an information system. Information bearers are ***about*** something else, while non-referring particulars are not about something else. Information bearers can be about not only non-referring particulars, an example being the driving license card of a person which is about its driving rights, but also about other information bearers, an example being a textual description of a specific person's driving license, stating, for instance, that the name of the driver is almost not readable. A copy of such a driving license can be at the same time about both the card and the rights enjoyed by the license holder.

4.1. Relations between information bearers and portions of reality

RT distinguishes explicitly and formally between various relations that obtain between information bearers and the various types of portions of reality it is capable of describing. These relations are:

- ***is-about***, which obtains between an information bearer and a portion of reality, such as, for example, a book about George W. Bush Sr. (the book being an information bearer) being about parts of the life of George W. Bush Sr. and his environment (a combination of several configurations in which figure, besides George W. Bush Sr., various other entities such as his advisors, friends, trips, speeches, and so forth).
- ***corresponds-to***, which obtains between an RT-tuple and a configuration;
- ***represents***, which obtains between a specific subtype of information bearer, namely what we call a '*representation*', and some further entity (or collection of entities). A representation is thus such that (1) the information it contains is about an entity, and not a configuration, external to the representation and (2) it stands for or represents that entity. Examples are an image, record, description or map of the United States. Note that a representation (e.g. a description such as 'the man over there on the corner') represents a given entity even though it leaves out many aspects of its target.
- ***denotes***, which obtains between data-elements expressed by means of a data type that we call '*denotator*' (see further) and an entity.
- ***contains***, which obtains between information bearers and can be used to express what pieces of information of a specific data type are parts of other pieces of information. An example is a digital message which contains RT-tuples describing configurations of entities in which a specific person figures.

4.2. Denotators

A denotator is a representational unit which denotes directly an entity in its entirety without providing a description. An example of a denotator is the string 'Bush' in the sentence 'President Bush visited Europe several times' when, whether or not known to the reader of the sentence in question, the writer had in mind a particular Bush, whether George Bush Jr. or George Bush Sr. The sentence itself is an information bearer according to our terminology. Because many representations are built out of constituent sub-representations as their parts, in the way in which paragraphs are built out of sentences and sentences out of words, RT uses the data type called '*representational unit*' to represent such smallest part. Examples are: icons, names, simple word forms, or the sorts of alphanumeric identifiers found in digital records. Note that many images are not composite representations since they are not built out of smallest representational units in the way in which molecules are built out of atoms (Pixels are not representational units in the sense defined.) [18].

RT distinguishes explicitly and formally between three types of denotators, referred to respectively as '*IUI*', '*UUI*' and '*CUI*'.

An IUI – abbreviation for '*Instance Unique Identifier*' – is a denotator in the form of a persistent, globally unique and singular identifier which denotes (or is believed to denote) a particular and which is managed in a referent tracking system. A UUI – for '*Universal Unique Identifier*' is a denotator which denotes a universal within the context of a realism-based ontology. A CUI – abbreviation for '*Concept Unique Identifier*' – is a denotator for entities of a type that is commonly and ambiguously called a '*concept*' [17], but which in BFO is called a '*defined class*', and defined as a subset of the extension of a universal which is such that the members of this subset exhibit an additional property which is (a) not shared by all instances of the universal, and (b) also might be exhibited by particulars which are not instances of that universal.

5. Referent Tracking System

A referent tracking system (RTS) is a special kind of digital information system which keeps track of (1) what is the case in reality and (2) what is expressed in other information systems about what is believed to be the case in reality. It does this unambiguously by means of the data types just sketched – in the first place resorting to IUIs – using principles and methods that assure – modulo the occurrence of errors, the resolution of which is also covered by the RT paradigm – that an IUI is (1) persistent because once created in a RTS it is never deleted, (2) globally unique because an IUI denotes only one entity within an RTS, and (3) singular because within an RTS, there is only one IUI for a specific entity.

Figure 2 shows the various components of an RTS and how an RTS can be used in association with external information systems and terminology (or ontology) servers. The direction of the arrows depicted therein shows the processing of service requests, the communication, however, being bi-directional to accommodate responses to the requests.

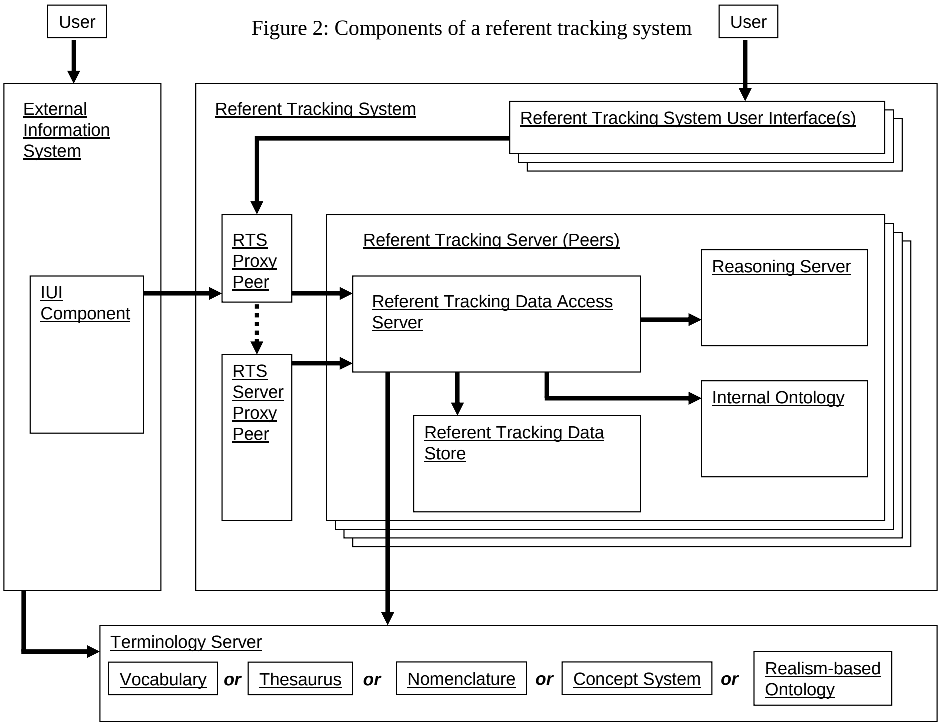

Figure 2: Components of a referent tracking system

5.1. Components of a referent tracking system

An RTS includes at least four types of components: (1) one or more referent tracking servers, (2) one or more referent tracking system user interfaces, (3) an RTS Proxy Peer, and (4) an RTS Server Proxy Peer. The components execute on one or more processors, computers or computing devices. Further, all of the components of an RTS can run on one computing unit; one or more components can run on one computing unit, while others run on one or more other computing units; or the components may be distributed among various computing units.

Each referent tracking server includes a data access server [24], which manages service requests coming from an RTS Proxy Peer or RTS Server Proxy Peer and which performs data manipulation on the server's main component: a referent tracking data store thereby assisted by a reasoning server. The latter performs various sorts of reasoning functions by combining data from the data store with information coming from external terminology servers. The type of reasoning that can be performed depends on whether the terminology server contains nomenclatures, vocabularies, thesauri, and so forth. The referent tracking server comes also with an internal ontology which is a repository dedicated, for instance, to store information obtained during the initialization process, access control information about authorized users and usages, and so forth. The referent tracking system user interfaces allow direct users of the RTS to perform (1) a variety of management functions such as registering new external information systems, configuring a referent tracking server, adding additional referent tracking servers, and so forth, and (2) content functions such as running patternmatching algorithms on the data in the referent tracking data store to detect inconsistencies, invoke triggers and alerts, perform population-based studies, and so forth.

5.2. Layered architecture

Figure 3 provides further details regarding the four-layered architecture of a RTS. The outer layer is a client side layer which connects to a RTS client which is typically a third party information system or a middleware component. The latter send a query to a Proxy Peer in the network layer that forwards the request to the appropriate RTS server in the network. During execution of the query, the RTS server calls the services of the RTS core API to retrieve the results from the Database Management System databases (DBMS) that constitute the data source layer.

A referent tracking data store includes, for instance, two parts: an *IUI-repository* and a *referent tracking database* (RTDB). The IUI-repository includes, as explained in section 6, the A-tuples and D-tuples which provide meta-information about information about first-order entities. The IUI-repository thus manages the statements about the assignment of IUIs to particulars, and provides a central repository of IUIs to the RTS. The RTDB is a database of statements representing the detailed information about particulars, examples being '#IUI-1 instantiates the universal Person' and '#IUI-1 has the name 'John''. The RTS Core layer implements the business logic of RT, namely, the insertion and retrieval of RT-tuples in any of its databases.

The IUI-repository and RTDB components are implemented through a series of application programming interfaces (APIs). The IUI-repository includes services to search particular representations and to insert new ones in its corresponding DBMS. Similarly, the RTDB components provide API get methods to search and create methods to insert tuples in its database.

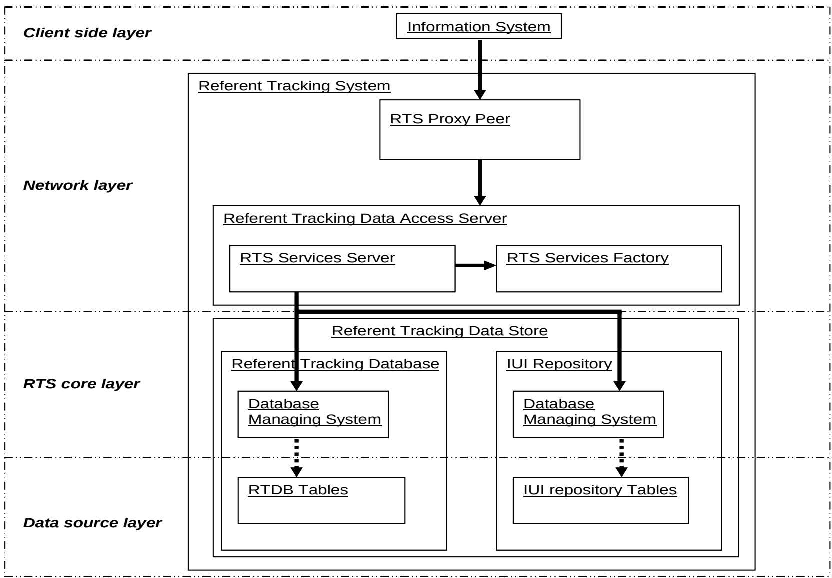

Figure 3: Layered implementation of a referent tracking system

The IUI-repository and RTDB components are implemented independently of any specific DBMS (e.g. MYSQL, HSQL). DBMS support is controlled by DBMS specific driver components, such as for MYSQL and HSQL.

Insertion services allow inserting a new RT tuple into the repository. The RT-tuples are inserted in a transaction, which is an information unit. As an example, entering a patient's blood pressure could involve a couple of RT statements which could include one or more RT-tuples. All tuples in a transaction are guaranteed to be committed in the data store. In case where either a system breaks down (by power failure or other means) or a user aborts the operation (e.g. a user closes/cancels the data entry screen while entering data), no partial information is stored in the data store. This service marks the start of a transaction for a specific session of a user. The RT paradigm does not allow any deletion operation in order to be able to always return to a state of the database as it was at a certain time in history. To prevent mistakes in creating new tuples in the IUI-repository, the tuples are cached right after the create operation. The client can remove or modify the tuples from the cache, as long as the commit service has not been called.

5.3. Networks of Referent Tracking Systems

Since referent tracking is to make reference to entities in reality by means of singular and globally unique identifiers, an ideal setup is one in which only one RTS is used worldwide. More realistic, however, is the adoption of the RT paradigm in a step-wise fashion: each organization first installs its own RTS, and afterwards connects them in expanding networks.

To support this evolution, as shown in , the RTS is built upon Peer to Peer (P2P) technology, enabling data sharing in such a way that a search query can be executed concurrently over distributed RTS servers (peers). In an RTS P2P network, a client thus sends a query to an RTS server which besides executing the query itself can forward it to other connected RTS servers for subsequent execution. Each peer then collects the results and sends them to the requesting peer. Finally, the RTS server who received the initial request returns the aggregated results to the client. Furthermore, an RTS P2P application is capable of database load sharing over multiple RTS server peers such that the network behaves as a singular database. This capability is useful in cases where a very large database cannot be hosted on a single machine, for instance because of computational limits. It includes also capabilities for discovering a new peer in a network, for authenticating users, and for ensuring secure communication.

shows an example of an RTS network in which three organizations, A, B and C, are running their own RTS peers. The peers are installed so that they are not directly known outside their corresponding organization's environment. In organization A, the Server Peers are alike in all respects and implement the objective of distributing a very large database load. When Information System A sends a search query to the RTS Proxy Peer within organization A, the latter forwards the query to all available Server Peers (A1, A2, …) in the organization which concurrently execute the query and return the results to the Proxy Peer that finally sends the results to the Information System. Each organization can form its own local group of servers whose membership is not known outside the organization. This protects against unauthorized access to the peers in the group. Controlled public access to each organization's data is offered through the Proxy Server peers. The separation of local peer advertisement within an organization from public (outside the host organization) contexts is the basis for the

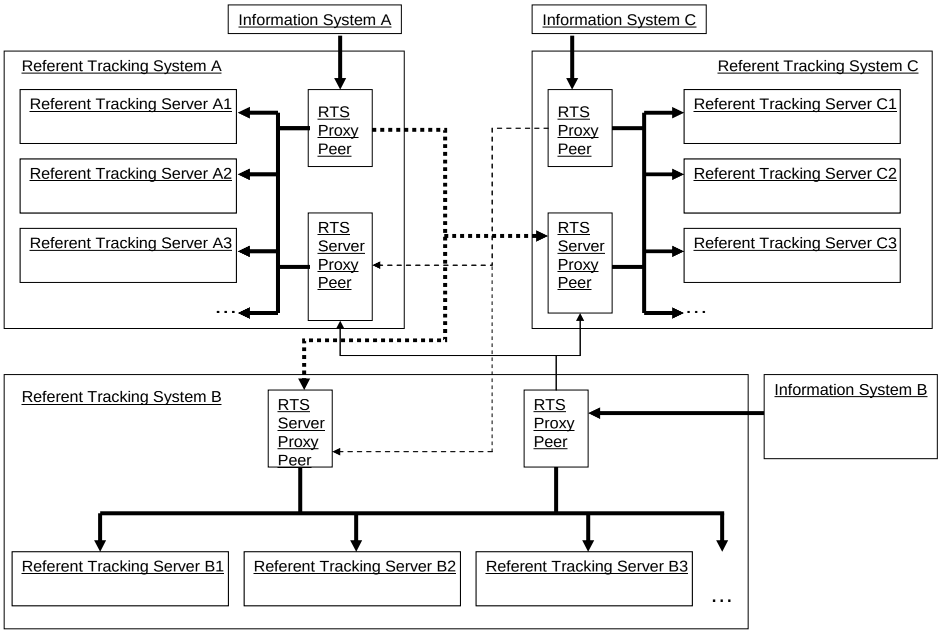

Figure 4: Peer-to-Peer implementation of Referent Tracking Systems

implemented security layer. The peers which are known locally provide full access to the local database, and the peers which are known publicly provide very restricted access to the database (they might, for instance, allow only searches over certain sorts of RT-tuples as explained further).

5.4. Reasoning services

Reasoning is a part of the RTS and its purpose is double. The first one is to prevent inconsistent data from being entered. By 'inconsistent data', we mean here data that cannot be true at the same time under the ontologies in whose terms the data are expressed. It is of course plausible that some analysts might be under the impression that, say 'John is in Paris' while others think that 'John is in London'. That analysts think different things is not inconsistent, but clearly they cannot both be right.

The second purpose for having reasoning services is to draw inferences during the execution of the search queries using the generic knowledge expressed in the terminology and ontology servers used to annotate the data and by exploiting the reasoners that operate on them.

Various third party reasoners exist, some being specific to a particular knowledge source, some coming with a public DIG (Description Logic Implementation Group) interface for description logic representations, while others use directly OWL-DL (Web Ontology Language-Description Logics).

In order to be able to deal with terminology servers and the various sorts of knowledge sources they offer (nomenclatures, thesauri, ontologies, ...), the RTS includes a Reasoning API which helps in sending reasoning queries uniformly to different terminology servers. The Reasoning API has an abstract class called OntologyConnector, which provides an interface to the external terminology systems by means of services. The interpretations of the OntologyConnector services are specific to a particular terminology server; therefore, a separate implementation of the OntologyConnector is required for each terminology server which is used to annotate the particulars in the RTS.

Description logics are widely used for building ontologies. The reasoners for such ontologies may take from 1 second to a day to compute inferences over the ontology classes depending on their size and definitional complexity. Therefore, instead of always directly communicating with the reasoners for each ontology when a specific query is launched, the RTS is able to store these queries and the results that have been returned by these reasoners as an inference graph in a database [24]. Thus, because the execution time of the OntologyConnector services can range from milliseconds to minutes depending on the query execution time in the external terminology system, the OntologyConnector caches the results returned from these systems. The cache is stored, for instance, in a RDBMS. During the execution of any of the OntologyConnector services, it first searches in the cache.

6. Referent Tracking Data Elements: RT-tuples

RT-tuples, although all corresponding to portions of reality, come in various flavors depending on the sort of information they contain.

6.1. A-tuples

A-tuples correspond to the assignment by some agent of an IUI to a particular. For the typical case, that particular is a pure first-order entity such as a specific person or a specific building about which information is to be stored in the RT system. However, by storing tuples, the RT system itself acts as an agent that assigns IUIs to the tuples itself. Indeed, for each insertion of an A-tuple, there is a corresponding insertion of a D-tuple that contains information about the corresponding A-tuple. To prevent infinite regress, the assignment of these IUIs does not involve the generation of an additional A-tuple, but is implemented through the use of these tuple-IUIs as an internal annotation to the tuple itself.

Three factors can be distinguished as structural elements involved in such an assignment act: (1) the generation of the relevant alphanumeric string, (2) its attachment to the relevant object, and (3) the publication of this attachment [20].

A-tuples are of the form $< IUI_p, IUI_a, t_{ap} >$ where IUI_p is the IUI of the particular in question, IUI_a is the IUI of the author of the assignment act, and t_{ap} is a time-stamp indicating when the assignment was made.

6.2. D-tuples

In light of the need or desire to resolve mistakes [25], RT includes the use of D-tuples, which are to be created whenever (1) a tuple other than a D-tuple is added to the RTS Data Store, in which case it includes meta-data about by whom and at what time the corresponding tuple was deposited or (2) a tuple, including D-tuples, is declared invalid in the system, in which case it includes additional info concerning the type of mistake committed and the reason therefore.

D-tuples are of the form $< IUI_d, IUI_T, t_d, E, C, S >$, where:

- IUI_T is the IUI of the tuple about which the D-tuple contains information.
- IUI_d: is the IUI of the entity annotating IUI_T by means of this D-tuple,
- *E* is either the symbol ‘I’ (for insertion) or any of the error type symbols as discussed further,
- *C* is a symbol for the applicable reason for change as discussed further,
- t_d is the time the tuple denoted by IUI_T is inserted or ‘retired’, and
- *S* is a list of IUIs denoting the tuples, if any, that replace the retired one.

6.3. PtoP-tuples

Descriptions which express configurations amongst particulars have the form of *PtoP* – particular to particular – tuples. Here again a number of structural elements can be distinguished: (1) an authorized user observes one or more objects which have already been assigned IUIs in the referent tracking system (RTS) in hand, (2) the user recognizes or apprehends that these objects stand in a certain relation, which is represented in some realism-based ontology, (3) the user asserts that this relation holds and publishes this assertion by entering corresponding data which are then published in the referent tracking data store.

This relationship data will then take the form of an ordered sextuple $<IUI_a, t_a, r, IUI_o, P, t_r>$, where

- IUI_a is the IUI of the author asserting that the relationship referred to by *r* holds between the particulars referred to by the IUIs listed in *P*;
- t_a is a time-stamp indicating when the assertion was made;
- *r* is the denotator in IUI_o of the relationship obtaining between the particulars referred to in *P*;
- IUI_o is the IUI of the ontology from which *r* is taken;
- *P* is an ordered list of IUIs referring to the particulars between which *r* obtains; and
- t_r is a time-stamp representing the time at which the relationship was observed to obtain.

P contains as many IUIs as are required by the arity of the relation *r*. In most cases, *P* will be an ordered pair which is such that *r* obtains between the particulars represented by its first and second IUIs when taken in this order.

6.4. PtoU-tuples

Another type of information that can be provided about a particular concerns what universal within an ontology it instantiates. Here, too, time is relevant, since a particular, through development, growth or other changes, may cease to instantiate one universal and start to instantiate another: thus George W. Bush Sr. changed from *foetus* to *newborn*, and from *child* to *adult*. Descriptions of this type (which we will refer to as *PtoU*-tuples – for: particular to universal) are represented by ordered tuples of the form $<IUI_a, t_a, inst, IUI_o, IUI_p, UUI, t_r>$, where

- IUI_a is the IUI of the author asserting that IUI_p is an instance (inst) of *UUI*;
- t_a is a time-stamp indicating when the assertion was made;

- *inst* is the denotator in IUI_o of the relationship of instantiation;
- IUI_o is the IUI of the realism-based ontology from which *inst* and *UUI* are taken;
- IUI_p is the IUI referring to the particular whose inst relationship with the universal denoted by UUI is asserted;
- *UUI* is the denotator of the universal in IUI_o with which IUI_p enjoys the inst relationship; and
- t_r is a time-stamp representing the time at which the relationship was observed to obtain.

Note that it is specified from which ontology inst and *UUI* are taken (and precisely which inst relationship in those cases where an ontology contains several variants). Such specifications not only ensure that the corresponding definitions can be accessed automatically, but also facilitate reasoning in the RTS Reasoning Server across ontologies that are interoperable with the ontology specified.

6.5. PtoC-tuples

Whereas for PtoU-tuples their denotators of relationships and universals are taken from realism-based ontologies rather than from other knowledge repositories in terminology servers, PtoC-tuples do allow CUIs to be used instead of UUIs. Of course, the relationship to be used is not to be some variant of 'inst' since the standard definitions in use for '*concept*' (such as '*unit of knowledge*' or '*unit of thought*') disallow most particulars from being declared as instances of concepts. PtoC-tuples (for particular to concept code) have the form $<IUI_a, t_a, IUI_c, IUI_p, CUI, t_r>$, where:

- IUI_a is the IUI of the author asserting that terms associated to *CUI* may be used to describe IUI_p;
- t_a is a time-stamp indicating when the assertion was made;
- IUI_c is the IUI of the concept-based system from which *CUI* is taken;
- IUI_p is the IUI referring to the particular which the author associates with *CUI*;
- *CUI* is the CUI in the concept-system referred to by IUI_c which the author associates with IUI_p; and
- t_r is a time-stamp representing a time at which the author considers the association appropriate.

Such tuples are to be interpreted as providing a facility equivalent to a simple index of terms in a work of scientific literature.

6.6. PtoU(-) – tuples

Since the RT paradigm requires that only entities that exist or have existed are to be assigned an IUI, a capability is provided that deals with what is called 'negative findings' or 'negative observations' as captured in expressions such as: 'no criminal history', 'membership of terrorist organization ruled out', 'absence of imminent danger', and 'attack prevented'. Such statements seem at first sight to present a problem for the referent tracking paradigm, since they imply that there are no entities in reality to which appropriate unique identifiers could be assigned. We therefore defined the relationship '*p* lacks *u* with respect to *r* at time *t*' such that there obtains a relation between the particular *p* and the universal *u* at time *t*, which is such that *p* stands to no instance of *u* in the relationship *r* at *t* [26, 27].

This ontological relation can be expressed by means of a 'PtoU(-) tuple' which is a lacks-counterpart of the PtoU-tuple and has the form <IUI_a, t_a, r, IUI_o, IUI_p, UUI, t_r>, expressing that the particular referred to by IUI_a asserts at time t_a that the relation r of ontology IUI_o does not obtain at time t_r between the particular referred to by IUI_p and any of the instances of the universal UUI at time t_r.

6.7. PtoN-tuples

Important particulars such as persons, ships, hurricanes, and so forth are often given proper names which function as denotators in reality outside the context of a referent tracking system. This sort of information is stored in an RTS by means of one or more 'PtoN-tuples' where 'N' stands for 'name'. These tuples have the form < IUI_a, t_a, nt, n, IUI_p, t_r , IUI_c >, where

- IUI_a is the IUI of the author asserting that n is a name of type nt used by IUI_c to denote IUI_p;
- t_a is a time-stamp indicating when the assertion was made;
- IUI_c is the IUI for the particular that uses the name n (this can be a person, a community of persons, an organization, an information system, ...);
- IUI_p is the IUI referring to the particular which the author associates with n;
- n is the name which the author associates with IUI_p;
- nt is the nametype (examples being first name, last name, nick name, social security number, and so forth); and
- t_r is a time-stamp representing a time at which the author considers the association appropriate.

7. Discussion

7.1. Referent Tracking and action-oriented formalisms

RT, at first sight, might look similar to other approaches. For instance, the need to track objects through time as they change, and to reason (and to have machines sometimes reason) over information that describes such changes, is what motivated calculi such as the situation calculus, the event calculus, and the fluent calculus, as well as some Knowledge Representation and Reasoning Systems. These approaches seek an efficient solution to the *projection problem* [28]: given an action theory that specifies the preconditions and effects of actions (including sensing), and a knowledge base about the initial state of the world, determine whether or not some condition holds after a given sequence of actions has been performed [29].

The situation calculus is a logic formalism that was first introduced by John McCarthy in 1963 [30] and since then underwent a few modifications [31]. The basic elements of situation calculus are: (1) *actions* that can be performed in the world, (2) *fluents* that describe the state of the world, each fluent thus being the representation of some property, and (3) *situations*. McCarthy and Hayes considered a situation to be '*a complete state of the universe at an instant of time*' [32], a position which is also maintained in fluent calculus [33], whereas others redefined situations as finite sequences of actions, thus a history of actions [31]. Event calculus does without situations, and uses only actions and fluents, whereby the latter are functions – rather than predicates as is the case in situation calculus – which can be used in predicates such as *HoldsAt* to state at what time which fluents hold [34].

RT differs in substantial ways from these logical formalisms. First of all, the goal of RT is not just to represent actions and changes, but all entities that exist in reality. Furthermore, these sorts of logics focus on computational aspects, but do not provide an integrated ontological characterization of entities such as actions, plans, and, because of their four-dimensionalist nature, for sure not of objects. It has been shown that it pays off to add more ontological rigor to formalisms such as situation calculus, for instance by using it only as one component for causal reasoning within a more elaborate, multi-component system [35].

RT, in contrast, is not in the first place a computational framework, but rather a representational one anchored in the realist view adhered to in Basic Formal Ontology (BFO) [23]. BFO distinguishes, for instance, continuants (such as George W. Bush) from occurrents (such as George W. Bush's life or his last trip from Washington to New York). These distinctions, including BFO's treatment of locations, positions and location schemes, was deemed essential in building a robot navigation model on top of situation calculus as embedded in Kuipers' Spatial Semantic Hierarchy [36]. Relationships of the sort expressed by, for instance, RT's PtoP- and PtoU-tuples hold only during certain time-periods [37, 38], and when they hold is expressed in the corresponding tuples themselves. In addition, PtoU-tuples express what universals a particular instantiates, thus also whether the entity described is an action or an object. Although no attempt has been made thus far, it seems plausible to assume that it is possible to express part of an RT database in terms of situation or event calculus.

7.2. Facts versus beliefs

The requirements within RT that tuples must make direct and explicit reference to that what they are about, and that this can only be done for entities that exist or have existed, would seem to make it very difficult to represent uncertain, or possibly deceptive knowledge. One can wonder if, for example, an intercepted communication contains 'Cain will strike down Abel' and it is believed that 'Cain' and 'Abel' are code words for non-personal entities, whether this belief can be recorded in this system. Similar questions can be asked about things in the future: isn't it important for a representational framework to be able to state knowledge about future happenings and entities that might not exist until the future, such as tomorrow's sunset or Al-Qaeda's next attack?

It is here that the distinction between three levels of reality as discussed in section 3.1 and the assignment of IUIs to RT-tuples themselves play a role. If a PtoP-tuple to which IUI-457 is assigned states that George W. Bush was president of the US in 2007, then the latter is taken to be a representation of reality – which of course may be a mistake – whereas IUI-457 is the proposition that the latter is the case. That this proposition is entertained (or not) by a specific person can be expressed by additional PtoP-tuples that relate the tuple in question to that person by referring also to adequate belief-related relations or processes depending on what sort of ontology is used. As in the case of action logics, RT itself does not come with a logic of beliefs, but from the representations, so we believe, secondary representations in terms of a belief logic can be generated.

For entities in the future, RT offers the possibility to *reserve* IUIs, rather than to *assign* IUIs [20]. Thus it is possible to assign an IUI to the plan to see and enjoy next

Sunday's sunset, whereas the detailed RT representation of that plan itself would contain a reserved IUI for that particular sunset.

7.3. Maintaining integrity

There are several challenges in maintaining the representational integrity of an RT system, specifically with respect to the requirements that an IUI within an RTS should denote only one entity, and that there is only one IUI for a specific entity. If, for instance, one doesn't know that 'Usama bin Ladin' and 'Osama bin Laden' denote the same individual, how could one possibly know to relate both names to the IUI denoting that individual? Here responsibility for faithful representation is shared between the user and the user interface. Whereas the former must devote enough effort to find out in each specific case what individual a name denotes, the latter, assisted by additional applications, must make it possible to reduce the effort required. Term comparison algorithms might be used to inform a user that a name similar to the one entered is already registered. Triggers and alerts can be implemented to warn a user that distinct individuals have the same name, and so forth. All this, however, does not guarantee that the right decision will be made in every case, and errors will very likely occur. So there have to be procedures to detect and correct mistakes. It is here that the D-tuples play an important role [25].

Easy to solve, once detected, are mistakes in which a particular has been assigned more than one IUI. In this case, only one of these IUIs would be used in future tuples, whereas all tuples in which the other IUIs are used will be replaced by tuples in which that one IUI will replace the redundant ones. This mechanism guarantees that it still remains known that during some period in the past, information concerning one particular was believed to be about two or more particulars.

More work would be required in the opposite case, i.e. when the same IUI is used to denote distinct particulars. Here it might be necessary to perform a manual revision of the tuples in which that is used.

To detect mistakes, the ontologies in whose terms RT-tuples are expressed can be used to guide integrity-checking routines that run over the RTDB. Because, for instance, persons (or any material continuant) cannot be at two distinct places in the same time, the presence of RT-tuples in the RTS that suggest this to be the case, indicates a mistake of the type 'one IUI for distinct particulars'. Logically, because two distinct material continuants cannot occupy the same spatial region, any collection of RT-tuples representing that this would be the case must contain an error of the type 'distinct IUIs for the same particular'.

7.4. RT and the Semantic Web

The various types of tuples enumerated in section 6 are expressible using standard Semantic Web technologies, though with some additional formalisms implemented at the data-base storage level. This is indeed the approach that has been taken in implementing the system [24].

The Resource Description Framework (RDF) [39] was used as the basic representation language. Our RDF representations of the RT-tuples are treated as resources themselves: each resource is therefore prefixed with the RTS name space URI and the prefix '*rts:*' such that, for instance, the resource *rts:IUI-1* is the same as *http://org.buffalo.edu/RTS#IUI-1*. To declare properties for resources, we used RDFS and mapped the RT-tuples to RDFS classes, thereby ensuring that the class names are identical to the template names, with the exception of PtoU-, which, because of restrictions in the RDFS naming conventions, has been mapped to *PtoLackU*.

Our implementation of the RTS is accessible through Web services which are invoked through SOAP messages [40] containing the procedure information (procedure name, parameters and return type) and port type (location of the procedure). The RTS uses Axis for Java [41] to host the web services thereby taking advantage of the native support of the Web Services Definition Language (WSDL) [42] that Axis provides.

The RTS has been build to be independent of any data source technology. To achieve this goal, we have defined the *RTRepository* class as an abstract Java class. This class provides all necessary services for managing the data based on the principles defined in the RT paradigm. To manage the RT data in a specific data source technology, an extension of the *RTRepository* for that specific technology is required. We have decided to develop the *RTRepositorySesameImp* class by extending the *RTRepository* such that it targets the SAIL Sesame API for manipulating RDF graphs as a data source [43].

Because the RT data are expressed in RDF, RDF query languages such as RQL [44], SPARQL [45] and SeRQL [43] can be used for retrieval. To this end, the *RTRepository* comes with the service '*repository.query(querystring, language)*' which has an argument for the query string and a second one for the name of the query language in which the first argument is expressed. The SeRQL query language is implemented with the help of the Sesame SeRQL query language module, and the SPARQL query language is implemented with the help of the ARQ query module (a SPARQL processor for Jena) [46].

8. Conclusion: meeting the new intelligence criteria

When set up in appropriate ways, a network of referent tracking systems is able to meet all the requirements identified for the envisioned Globally Networked and Integrated Intelligence Enterprise (see section 1.3).

The requirement to share intelligence data while still addressing the need to protect privacy, civil liberties, and sources and methods (C1), can be met by using the IUIs, typically the ones that stand proxy for persons, as pseudonyms. It would even be possible to go much further, for instance that all the information collected by credit card companies, banks, department stores, telecom providers and so forth would be pooled. Most citizens would find it unacceptable if that information were used for intelligence purposes without there being any reason to do so. But with the appropriate setup of IUIRepositories and RTDBs in such a way that, for instance, one specific agency has the means to link IUIs to persons, but otherwise no access to other RT-data, while other agencies would be able to do data-mining and pattern analysis on the pseudonymized data, no privacy or civil liberties would be violated. When analysts would detect suspicious patterns in the pseudonymized data pool, similar mechanisms as search warrants can be used to obtain re-identification of the data.

The requirements to provide services across agencies, partners, and international borders for multiple mission use (C2) and to be able to adapt rapidly to changing needs and new partners (C3) are supported by the possibility for referent tracking systems to cooperate in growing networks.

The C4 requirement, i.e. to have security built into the data and environment using tags, together with the C5-requirement that access should be based on attributes that go beyond security classification, is met by the specific ways in which RT-tuples are set up: they contain in every case an indicator for the provenance of the data and all data are coded by means of ontologies or terminologies. Furthermore, each RT-tuple can be treated as a first-order entity, thereby receiving its own IUI, and that IUI can be used in other RT-tuples, for instance to describe to what type of entities or specific entities it may be disclosed. The same IUI can be used to track the flow of the data-element throughout the intelligence network.

Data stewardship, finally, focusing on quality and reusability of data rather than, but not excluding, protection (C6) is a natural feature of the paradigm. One reason are the principles for IUI assignment which require that before an IUI is assigned to an entity, it should be checked whether that entity has already an IUI assigned to it. Mistakes will happen, of course, but they are traceable over time; if, for instance, when data accumulate, two IUIs start to appear repeatedly in the same configuration, then they may stand proxy for the same entity. Or, if the database at some stage contains a PtoP-tuple stating that the entity with IUI_x was in some place at a given point in time, while in a completely different place a bit later, then it is likely, modulo other types of mistakes, that IUI_x is denoting different things.

A problem, at first sight, might be the amount of work required to represent information in this way. But here again, other types of software such as natural language processing applications, might assist. Furthermore, as shown in [47, 48], it is in many cases possible to translate structured information into a form that is RT-compatible automatically. We argue that the effort to make systems of this kind acceptable is not greater than the effort to bring about the change in mindset to realize Vision 2015.

9. References

[1] Central Intelligence Agency. What is Intelligence? 2007 June 20, 2008 [cited 2008 August 12]; Available from: https://www.cia.gov/news-information/featured-story-archive/2007-featured-story-archive/what-is-intelligence.html

[2] Reagan R. Executive Order 12333--United States intelligence activities. 1981.

[3] United States Intelligence Community. The Intelligence Process. 2008 [cited 2008 August 12]; Available from: http://www.intelligence.gov/2-business.shtml

[4] Travers R. A Blueprint for Survival; The Coming Intelligence Failure. Studies in Intelligence. 1997;Semiannual Edition, No. 1:35-43.

[5] Chen H. Intelligence and Security Informatics for International Security. Information Sharing and Data Mining. New York: Springer-Verlag 2006.

[6] Office of the Director of National Intelligence. United States Intelligence Community Information Sharing Strategy. 2008.

[7] Hasselbring W. Information system integration. Communications of the ACM. 2000;43(6):33-8.

[8] Office of Homeland Security. National Strategy for Homeland Security. Washington D.C.: Office of Homeland Security 2002.

[9] H. Chen, R. Miranda, D. Zeng, T. Madhusudan, C. Demchak, Schroeder J. Intelligence and Security Informatics: Proceedings of the First Symposium on Intelligence and Security Informatics (ISI'03). New York: Springer-Verlag 2003.

[10] Chen H, Wang F-Y, Zeng D. Intelligence and Security Informatics for Homeland Security: Information, Communication, and Transportation. IEEE Transactions on Intelligent Transportation Systems. 2004 December;5(4):329-41.

[11] Wang GA, Atabakhsh H, Petersen T, Chen H. Discovering Identity Problems: A Case Study. *IEEE international conference on intelligence and security informatics*. Atlanta, GA, USA Springer 2005.

[12] Cybenko G, Giani A, Thompson: P. Cognitive Hacking: A Battle for the Mind. IEEE Computer. 2002;35(8):50-6.
[13] Mercado S. A Venerable Source in a New Era: Sailing the Sea of OSINT in the Information Age. CIA Studies in Intelligence. 2004;48(3):45-55.
[14] Office of the Director of National Intelligence. Office of the Director of National Intelligence. 2008 August 22 [cited 2008 August 28]; Available from: http://www.dni.gov/index.html
[15] Office of the Director of National Intelligence. Vision 2015: A Globally Networked and Integrated Intelligence Enterprise. 2008.
[16] Office of the Director of National Intelligence. The National Intelligence Strategy of The United States of America. Office of the Director of National Intelligence 2005.
[17] Smith B. Beyond concepts: ontology as reality representation. *Proceedings of the third international conference on formal ontology in information systems (FOIS 2004)*. Amsterdam: IOS Press 2004:73-84.
[18] Smith B, Kusnierczyk W, Schober D, Ceusters W. Towards a Reference Terminology for Ontology Research and Development in the Biomedical Domain. *KR-MED 2006, Biomedical Ontology in Action*. Baltimore MD, USA 2006.
[19] Bittner T, Smith B. A Theory of Granular Partitions. In: Duckham M, Goodchild MF, Worboy MF, eds. *Foundations of Geographic Information Science*. London: Taylor & Francis Books 2003:117-51.
[20] Ceusters W, Smith B. Referent Tracking in Electronic Healthcare Records. In: Engelbrecht R, Geissbuhler A, Lovis C, Mihalas G, eds. *Connecting Medical Informatics and Bio-Informatics Medical Informatics Europe 2005*. Amsterdam: IOS Press 2005:71-6.
[21] Ceusters W, Smith B. Referent Tracking for Digital Rights Management. International Journal of Metadata, Semantics and Ontologies. 2007;2(1):45-53.
[22] Ceusters W, Smith B. Referent Tracking for Corporate Memories. In: Rittgen P, ed. *Handbook of Ontologies for Business Interaction*. New York and London: Idea Group Publishing 2007:34-46.
[23] Grenon P, Smith B, Goldberg L. Biodynamic Ontology: Applying BFO in the Biomedical Domain. In: Pisanelli DM, ed. *Ontologies in Medicine*. Amsterdam: IOS Press 2004:20-38.
[24] Manzoor S, Ceusters W, Rudnicki R. Implementation of a Referent Tracking System. International Journal of Healthcare Information Systems and Informatics. 2007;2(4):41-58.
[25] Ceusters W. Dealing with Mistakes in a Referent Tracking System. In: KS H, ed. *Proceedings of Ontology for the Intelligence Community 2007 (OIC-2007)*. Columbia MA 2007:5-8.
[26] Ceusters W, Elkin P, Smith B. Referent Tracking: The Problem of Negative Findings. In: Hasman A, Haux R, Lei Jvd, Clercq ED, Roger-France F, eds. *Studies in Health Technology and Informatics Ubiquity: Technologies for Better Health in Aging Societies - Proceedings of MIE2006*. Amsterdam: IOS Press 2006:741-6.
[27] Ceusters W, Elkin P, Smith B. Negative Findings in Electronic Health Records and Biomedical Ontologies: A Realist Approach. International Journal of Medical Informatics. 2007 March;76:326-33.
[28] Reiter R. Knowledge in Action. Logical Foundations for Specifying and Implementing Dynamical Systems. Boston: MIT Press 2001.
[29] Vassos S, Levesque H. Progression of Situation Calculus Action Theories with Incomplete Information. In: Veloso M, ed. *Proceedings of IJCAI-07* 2007.
[30] McCarthy J. Situations, actions and causal laws. Stanford, CA: Stanford University Artificial Intelligence Laboratory; 1963.
[31] Reiter R. The frame problem in the situation calculus: a simple solution (sometimes) and a completeness result for goal regression. In: Lifshitz V, ed. *Artificial intelligence and mathematical theory of computation: papers in honour of John McCarthy*. San Diego, CA, USA: Academic Press Professional, Inc 1991:359-80.
[32] McCarthy J, Hayes PJ. Some philosophical problems from the standpoint of artificial intelligence. Machine Intelligence. 1969;4:463-502.
[33] Thielscher M. Introduction to the Fluent Calculus. Electronic Transactions on Artificial Intelligence. 1998;2(3-4):179-92.
[34] Kowalski R. Database updates in the event calculus. Journal of Logic Programming. 1992;12(1-2):121-46.
[35] Kuipers B. The spatial semantic hierarchy. Artificial Intelligence. 2000 May;119(1-2):191 - 233.
[36] Bateman J, Farrar S. Modelling Models of Robot Navigation Using Formal Spatial Ontology. *Spatial Cognition IV Reasoning, Action, and Interaction*. Berlin / Heidelberg: Springer 2005:366-89.
[37] Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, Lomax J, et al. Relations in biomedical ontologies. Genome Biology. 2005;6(5):R46.
[38] Smith B, Grenon P. The Cornucopia of Formal-Ontological Relations. Dialectica. 2004;58(3):279-96.
[39] Manola F, Miller E. RDF Primer. 2004 [cited; Available from: http://www.w3.org/TR/rdf-primer/
[40] Mitra N. SOAP Version 1.2 Part 0: Primer. W3C Recommendation 2003.

[41] The Apache Software Foundation. Axis: A Webservices toolkit. 2005 [cited 25 January, 2007]; Available from: http://ws.apache.org/axis/

[42] Christensen E, Curbera F, Meredith G, Weerawarana S. Web Services Description Language (WSDL) 1.1. *W3C Note* 2001.

[43] Broekstra J, Kampman A, Harmelen Fv. Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema. *Lecture Notes in Computer Science - International Semantic Web Conference ISWC2002*. Heidelberg: Springer 2002:54-68.

[44] Foundation for Research and Technology – Hellas. The RDF Query Language (RQL). 2003 July 18 [cited 25 January 2007]; Available from: http://139.91.183.30:9090/RDF/RQL/

[45] Prud'hommeaux E, Seaborne A. SPARQL Query Language for RDF. W3C Working Draft 2006 October 4th [cited January 22, 2007]; Available from: http://www.w3.org/TR/rdf-sparql-query/

[46] RDF Data Access Working Group. ARQ - A SPARQL Processor for Jena. 2007 [cited 15th Febuary, 2007]; Available from: http://jena.sourceforge.net/ARQ/

[47] Rudnicki R, Ceusters W, Manzoor S, Smith B. What Particulars are Referred to in EHR Data? A Case Study in Integrating Referent Tracking into an Electronic Health Record Application. In: Teich JM, Suermondt J, C H, eds. *American Medical Informatics Association 2007 Annual Symposium Proceedings, Biomedical and Health Informatics: From Foundations to Applications to Policy*. Chicago, IL 2007:630-4.

[48] Manzoor S, Ceusters W, Rudnicki R. A Middleware Approach to Integrate Referent Tracking in EHR Systems. In: Teich JM, Suermondt J, C H, eds. *Proceedings of the American Medical Informatics Association 2007 Annual Symposium Biomedical and Health Informatics: From Foundations to Applications to Policy*. Chicago IL 2007:503-7.