Data Integration, Semantic Data Representation and Decision Support for Situational Awareness in Protection of Critical Assets

Atta Badii, Marco Tiemann and Daniel Thiemert
University of Reading, Whiteknights, Reading, RG6 6AH, U.K.

Keywords: Data Representation, Semantic Web, Decision Support, Security.

Abstract: This paper presents the design and development of a system for data integration, data representation, situational awareness and decision support that has been developed in the EC-co-funded research project MOSAIC. The paper motivates the architecture and describes the data representation model and the developed system components. It discusses the approach for improved situational awareness and decision support as a novel integration of systems developed under the MOSAIC project as deployed for the protection of critical assets as a demonstrator.

1 INTRODUCTION

The protection of critical assets is an important area of concern for public and private bodies that are responsible for this task. Critical assets may include fixed assets such as specific sites or buildings but also events of a temporary nature that increase the criticality of a location, for example, when large public gatherings need to be protected. While relevant organisations such as police forces fundamentally have a range of data sources with relevant information available, such data sources are in practice often not exploited in terms of the data that can be extracted from unstructured text or video data and are rarely connected with each other, so that users may need to access dozens of different systems manually in order to gather potentially relevant data on their personal desktops and then proceed to manually analyse the gathered data (similarly described in Smith et al., 2012).

The EU-co-funded MOSAIC project investigates and implements a system designed to support relevant bodies, primarily police organisations, in protecting critical assets. The focus of the project is on the analysis, integration and use of data collected from heterogeneous data sources such as existing databases, manually written intelligence reports and notes and Closed-Circuit TV (CCTV) video footage. Figure 1 depicts the main components of the MOSAIC system.

This short contribution describes the data representation, data storage, import and access and decision support components of the MOSAIC system. Data representation in MOSAIC provides a unifying semantic framework for the representation of all data available through the MOSAIC system by means of domain ontology. The Data storage and access system component comprise the technical infrastructure that manage the data provided to the individual system components. This infrastructure handles the integration of data that arrive in proprietary formats of individual data analytics components (e.g. ONVIF format video event data coming from video analytics components; ONVIF, 2014).

The decision support component evaluates available and newly incoming data in order to determine whether any action needs to be taken based on the latest operational picture as represented by the overall data model available at each point in time and to initiate actions in coordination with relevant staff such as intelligence analysts or CCTV system operators and supervisors.

2 DATA REPRESENTATION

Since the data to be represented in the MOSAIC data model should be accessible via a single point of access and should be accessible using a single mechanism or language, it is necessary to formulate

Badii A., Tiemann M. and Thiemert D..
Data Integration, Semantic Data Representation and Decision Support for Situational Awareness in Protection of Critical Assets.
DOI: 10.5220/0005126603410345
In *Proceedings of the 11th International Conference on Signal Processing and Multimedia Applications* (MUSESUAN-2014), pages 341-345
ISBN: 978-989-758-046-8

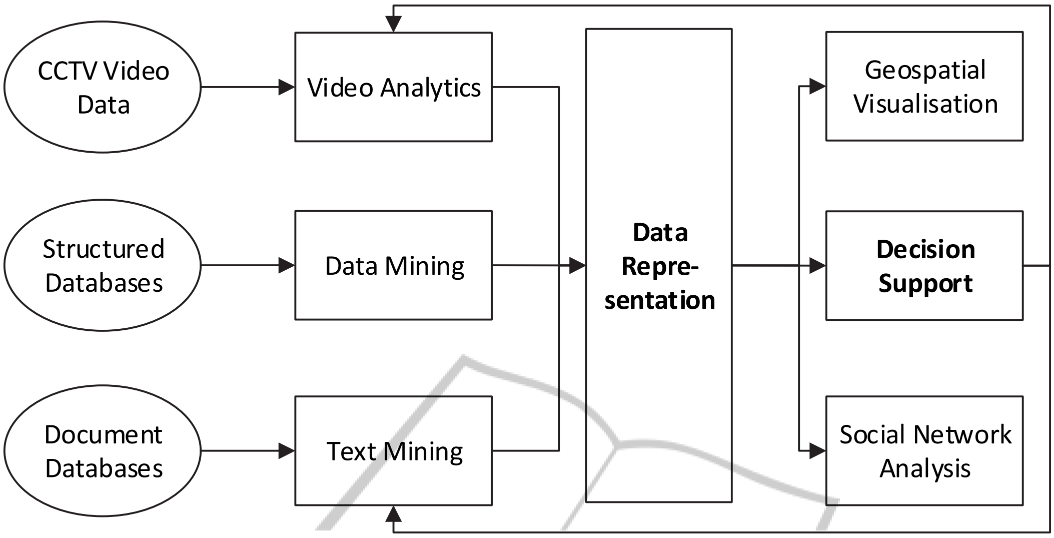

Figure 1: Overview of MOSAIC system components. Components considered in this paper are highlighted bold.

a single data representation format and a single overall data model to be used for this. While two of the data analysis components in the MOSAIC system use clearly specified textual description formats to formally describe their analysis outputs, neither of the two are suitable for representing data from all data sources relevant to MOSAIC, and the formats are not formalised suitably for advanced reasoning and decision support functionalities.

Hence, a dedicated data representation model has been developed in order to support the aforementioned requirements. This MOSAIC data model represents the domain using an ontology representation based on a primarily hierarchical domain description. This ontological model represents classes of prototypical entities in the model; observed data that fit into the ontology model are then added to the model as individuals. Table 1 lists the top-level entity types that are used in the MOSAIC data model ontology. A similar, but not event-oriented modelling approach has been described by Lee, (2007); an approach that focuses exclusively on event-based representation has been described by Snidaro et al., (2007).

Each of the top-level entity types in the model is further defined via distinctive entity types. Figure 2 shows the two hierarchical layers below the "Event" entity type as an illustration. The "is-a" relationships indicate hierarchical subordination.

In addition to the primarily hierarchical main entity types, general and entity-type-specific property models are also part of the domain model and establish relationships across hierarchies, in particular at the subcategory entity level.

Table 1: Top-Level elements of the MOSAIC hierarchical ontology model.

Name	Description
Actor	An actor can be a person or group that may potentially carry out actions relevant to the system data model
Object	An object can be any object or passive entity that is expected to never carry out any action (to be classified as an actor if it is)
Event	A specific occurrence that can be related to actor(s), place(s) and time(s) and may involve object(s)
Place	A physical or virtual location organised in a strictly hierarchical model where possible
Time	A representation of a time instants or time intervals, generally used to specific events
Metadata	Data concerning authorship, data provenance and access rights for data instances in the data mode

The MOSAIC domain model is formally structured and represented as an OWL-Lite ontology (McGuiness and van Harmelen, 2004) and predominantly uses the Resource Description Framework (RDF) data format in XML notation (RDF/XML) for data exchange. The selection of OWL-Lite and RDF/XML has been motivated by the widespread availability of suitable tools for development and data storage (see the following section) and the usage of the Web Ontology Language (OWL) as part of many Semantic Web applications, (van Ossenbruggen et al., 2004; Nack et al., 2005) because of which developers familiar with the latter should be able to connect to and

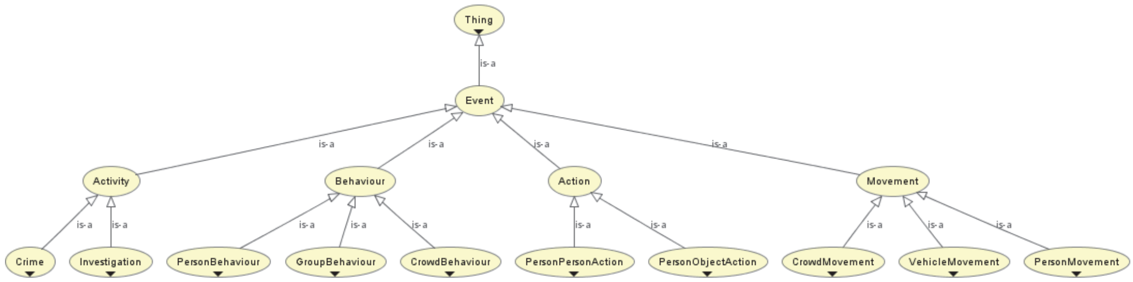

Figure 2: Simplified subset of event types used in MOSAIC, used for system demonstrations. The figure shows the middle layer of children of the event class, which are further specified with additional child nodes (not depicted due to space constraints).

develop based on MOSAIC-based systems with little additional learning effort.

3 DATA STORAGE, ACCESS AND INTEGRATION

The MOSAIC system requires a data store that as a minimum acts as a facade providing a single point of access to all data available to a user in the system. In order to facilitate this, a dedicated data store component has been implemented as part of the project. This data store acts as a facade for access to source data such as text data or analysis output that does not conform to the MOSAIC data model format, and stores all relevant data provided to it natively as data triples, so that the data model is represented via entities and relationships as a directed acyclic graph of subject – relationship – object.

The implemented system is based on the Apache Fuseki server system (Apache Fuseki, 2014), which itself contains an instance of the Apache Jena Semantic Web stack with an integrated triple store database system (Apache Jena, 2014). As a native Semantic Web stack, Apache Jena directly supports the use of the MOSAIC data model and uses the SPARQL data manipulation language for create, read, update, delete (CRUD) operations.

The system has been extended with specific functionalities required for use in the MOSAIC project, for example storing persistent queries in the system which then sends a notification when new data that matches the query is found. All functionalities of the MOSAIC data store are exposed via Web Services; Web Services are also used for data integration.

In terms of data integration, the data store needs to be able to process input from the MOSAIC data analysis components in so far as they do not communicate using the MOSAIC data model format. This is the case for Text Mining and Video Analytics components. Both communicate using specified XML document formats which do not conform to RDF; both formats use a limited vocabulary. The data integration (or semantic mediation) is carried out using an Extensible Stylesheet Language (XSL) transformation that converts the input formats into MOSAIC data model RDF input suitable for the data store.

4 SITUATIONAL AWARENESS AND DECISION SUPPORT

One of the main use cases for which a unified data representation is required is to enable automated data processing for decision support. In this contribution, the consideration of decision support functionalities focuses on reasoning systems for exploiting the expressiveness of the MOSAIC data model and for defining and processing production rules that are used to react to specific constellations of data that may be of interest to intelligence analysts or CCTV system operators. MOSAIC provides further decision support components which are not discussed in this paper.

Reasoner-based decision support in MOSAIC can be differentiated into two main tasks. First, a reasoner is needed in order to identify implicit and derived relations and properties within the MOSAIC domain model so that they can be used in standard queries and for decision support. MOSAIC uses a rule engine integrated into the Apache Jena Semantic Web stack to achieve this. Figure 3 shows a simple example of how the output of this reasoning process links entity instances that are part of an entity type hierarchy.

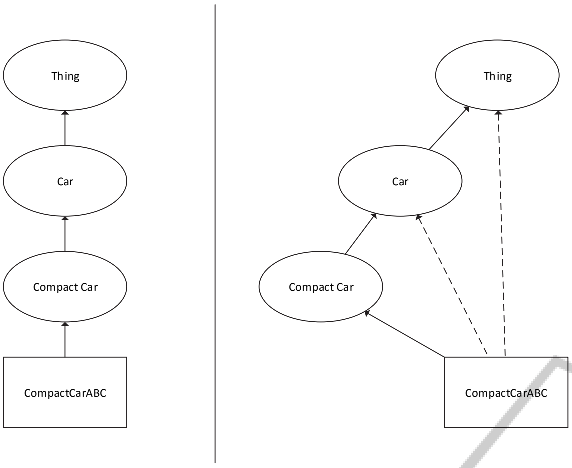

Figure 3: Comparison of relations that can be used in queries prior to applying an ontology reasoner (left) and after having applied an ontology reasoner (right). Oval elements indicate concepts, the rectangular element is an instance of the entity "CompactCar", all directed arrows indicate "is-a" subsumption relations, fully drawn lines indicate explicit relations and dotted lines indicate relations that have been made explicit through an ontology reasoner.

The reasoning functionality described above is essential for making the full power of the ontology model available to users of the data store (see Ulicny et al., 2008 for benefits and problems to be addressed in this context). However the production rule system implemented as part of the MOSAIC system allows users to automatically analyse data in the data store as these become available and to perform specific actions within the system when the left-hand-side "if" conditions of a production rule are satisfied.

To achieve this, the well-known JBoss Drools (JBoss, 2014) production rule engine has been integrated with the MOSAIC data store via a connector that allows it to access the data of the MOSAIC ontology data model and a custom data model representation of the MOSAIC data model for use with JBoss Drools has been implemented.

Drools is an efficient production rule processing engine for which the powerful Drools Rule Language has been developed. This language facilitates complex event processing such as rules with temporal order or other complex event conditions, which sets the functionalities of the rule engine apart from simpler custom production rule engines and from query languages such as SPARQL.

The Drools Rule Language divides rules into left-hand-side "if" and right-hand-side "then" parts; actions for the right-hand-side of a rule can be defined in the Java programming language and can include calls of custom methods, so that in principle any functionality that can be implemented in Java can be triggered via a Drools rule once implemented.

In MOSAIC, different notification functions that alert analysts or operators to interact with CCTV cameras (e.g. to turn a camera to a new position) and functionalities that add new information derived from reasoner output to the MOSAIC data store have been implemented. The code snippet given in Figure 4 shows the formulation for a simple example Drools rule that would fire and send an email once a specific vehicle has been spotted at a specific location.

Rules can be evaluated every time additional data is made available in the data store or be configured with cool down periods or number of times to fire in order to avoid excessive numbers of rule activations.

In combination with MOSAIC user interface components and map-based visualisation systems, the decision support system allows the combination of various complex situations that may be of interest to intelligence analysts or CCTV operators. In particular in combination with geospatial visualisation as described in another contribution submitted to this event (Badii et al., 2014), the decision support system can be a powerful aid when dealing with large amounts of simultaneously incoming data as is the case for instance for CCTV operators.

5 CONCLUSION

The work described in this paper describes the components of the MOSAIC system that are concerned with integrating data from heterogeneous data sources so that they can be accessed in a unified manner and with empowering users to define rules that reflect complex information needs as they may arise when protecting critical assets from a wide range of possible threats.

The solution described in this contribution furthermore shows how the currently prevalent problems of segregation of data in data silos and the subsequent need for large amounts of manual labour in navigating, collating and evaluating the gathered data can be supported and made more efficient for intelligence analysts and how personnel such as CCTV operators can be supported by intelligent integrated systems.

The work presented here is a foundation that is used in the MOSAIC system and it is anticipated that it may be used for future research on new

```
rule "Van sighted in Exampleville"
        when
                statement1:
                ModelStatement(predicate == "moscor:isEventConnectedToPlace",
                object == " moscor:County_Exampleville")
                statement2:
                ModelStatement(subject == statement1.getSubject(),
                predicate == "moscor:isEventConnectedToObject",
                object == "moscor:Van_AB12AAA_Ford_Escort")
        then
                mail.Email.send("Van sighted alert: " statement2.getObject()
                + " was sighted in " + statement1.getObject());
end
```

Figure 4: Example rule that notifies a user when a van sighting is reported in a geographical area.

advanced techniques for data analysis for the protection of critical assets.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 261776.

REFERENCES

Apache Fuseki, 2014. http://jena.apache.org/documentation/serving_data/.

Apache Jena, 2014. http://jena.apache.org.

Badii, A., Tiemann, M., Adderley, R., Seidler, R., Evangelio, R., Senst, T., Sikora, T., Panattoni, L., Raffaelli, M., Cappel-Porter, M., Husz, Z., Hecker, T., Peters, I. (2014). Proceedings of the SIGMAP 2014 *11th International Conference on Signal Processing and Multimedia Applications*, Vienna, Austria, 28-30 August, 2014.

JBoss Drools, 2014. http://drools.jboss.org.

Lee, R., 2007. The Use of Ontologies to Support Intelligence Analysis. In: *Ontology for the Intelligence Community (OIC-2007)*, November 28-29, 2007, Columbia, Maryland, USA.

McGuiness, D., van Harmelen, F., 2004. OWL Web Ontology Language Overview. W3C Recommendation 10 February 2004. http://www.w3.org/TR/owl-features/.

Nack, F., van Ossenbruggen, J., Hardman, L., 2005. That Obscure Object of Desire: Multimedia Metadata on the Web, *Part 2. IEEE Multimedia*, vol. 12, no. 1.

Open Network Video Interface Forum, 2014. http://www.onvif.org.

Smith, B., Malyute. T., Salmen, D., Mandrick, W., Parent, K., Bardhan, S., 2012. Ontology for the Intelligence Analyist. CrossTalk, November/December 2012, pp. 18-25.

Snidaro, L, Belluz, M., Foresti, G., 2007. Domain Knowledge for Surveillance Applications. In: Proc. *10th International Conference on Information Fusion, 9-12 July 2007*, Quebec, Canada.

van Ossenbruggen, J., Nack, F., Hardman, L., 2004. That Obscure Object of Desire: Multimedia Metadata on the Web, *Part 1. IEEE Multimedia, vol. 11, no. 4.*

Ulicny, B., Matheus, C., Kokar, M., Powell, G. (2008). Problems and Prospects for Formally Representing and Reasoning about Enemy Courses of Action. FUSION, p. 1-8.