Semantic Technologies in Electronic Government

Tutorial and Workshop

Steve Harris, Jeremy Gibbons, Jim Davies, and Charles Crichton
Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
http://www.comlab.ox.ac.uk/firstname.lastname/

ABSTRACT

Joined-up government depends fundamentally on semantics —- on the computable representation of meaning, so that data is associated with appropriate metadata from the start, and this association is maintained as the data is manipulated. This paper summaries a tutorial and workshop on semantic technologies for supporting electronic government.

Categories and Subject Descriptors

J.1 [**Administrative Data Processing**]: Government; H.2.3 [**Database Management**]: Languages—*Data description languages, data manipulation languages*; H.3.5 [**Information Storage and Retrieval**]: Online Information Services—*Data sharing, web-based services*; K.5.2 [**Legal Aspects of Computing**]: Governmental Issues—*Regulation*; I.2.4 [**Artifical Intelligence**]: Knowledge Representation Formalisms and Methods—*Relation systems, semantic networks*; K.6.2 [**Management of Computing and Information Systems**]: Security and Protection; K.4.1 [**Computers and Society**]: Public Policy Issues—*Transborder data flow*

General Terms

Standardization, legal aspects

Keywords

Metadata, semantic markup, interoperability, semantic web, ontology, application firewall, XML, XMI, RDF, SPARQL, OWL, SKOS, SAWSDL, ISO 11179.

1. INTRODUCTION

'Joined-up Government' — everyone wants it, but what is it? The phrase is attributed [20] to Tony Blair, then Prime Minister of the UK, at the time of the publication of Britain's first e-government strategy [16]. The term has become synonymous with the integration of services, processes, systems, data, and applications necessary to achieve a seamless, citizen-centered government [4, p1].

ICEGOV 2008, Cairo, Egypt

In the introduction to *Roadmapping eGovernment Research* [4], Codagnone and Wimmer note a number of problems in achieving the extensive implementation of ICT required for joined-up government, including 'rigid and ineffective internal and inter-institutional processes', 'a severe lack of understanding of citizens' real needs, attitudes and abilities', business processes 'not properly designed for effective implementation', together with existing heterogeneous, fragmented and non-interoperable systems. While there is obviously inefficiency in government, and its legacy systems cannot meet all of today's requirements, these issues are as much a failing of modern ICT approaches, technology, and capability and in the nature of the business of government as they are the failings of citizens, civil servants, and politicians who direct, design, and use the services.

The sheer scale, inclusivity, reliability, transparency, and diversity of government lies beyond current business computing capability. In Europe, Government is a business whose turnover is 49% of gross domestic product; re-engineering a business process to gain competitive advantage in one market sector at the expense of another is perfectly acceptable for a company, but a workflow that excludes a minority from accessing a government service in a democracy is not; data retention in business is focussed upon the value to future sales, whereas in government requirements are frequently absolute and permanent; Amazon sells things that have discrete value from suppliers who align themselves for a smooth transaction with one of the world's biggest retailers, but taxes, benefits, borrowing, subsidies, loans, and grants are far more diverse, fluid, conditional, and often given grudgingly or taken in excess. There is much that can be done better with current best practice — but only so much progress can be made. Citizen-centric government requires more from ICT than it is capable of supplying.

Possibly the fundamental requirement for the integration that lies at the heart of citizen-centric government is that collaborating services understand the meaning of the data they share. Put quite simply, as more information becomes available, it is increasingly unlikely that the authors of information will meet the software engineers developing solutions around it — much less the ultimate consumers of that information — so that they can explain the meaning and purpose of the data. Similarly, if we wish to implement information systems that operate in a wider range of circumstances without recourse to human intervention, then the machines must be able to access and act upon the meaning of the data as part of the logic of their operation. The failing of current business computing in the light of this latter requirement is particularly clear — by implication, we expect our services to encode government policy, yet that encoding is distributed throughout the program text and configuration of the information system, and entwined with code that addresses more prosaic concerns such as exception handling, file and record access, internal consistency, communication between tiers, and so on.

Data is held in a wide range of systems, including relational database management systems (RDBMS), XML databases, spreadsheets, documents, and webpages; the data may be widely distributed in physical and administrative terms, such as in coordinated child protection across healthcare, social services, and police, or in government-to-government communication of passenger name records; or the data might be required many years after its recording. The latter requirement is particularly onerous: in the UK, many public records are to be '*permanently* preserved and for their safe-keeping' (UK Public Records Act 1958). (Coincidentally, it was also during 1958 that integrated circuits — the key breakthrough that has lead to the modern information age — were first demonstrated by Kilby [17] and Noyce [23].) Processing occurs in a wide variety of computing languages, distributed logically and often physically between application components. The philosophy and capabilities of the languages differ widely between the object orientation of Java and C#, the data orientation of SQL and XQuery, and the procedural orientation of scripting languages.

In 'joined-up government', a record of birth would be input on a computer in a hospital, communicated through to a national register of births and deaths, become integrated into a chip embedded within a passport, and finally be read and displayed in another (human) language in another date format on another computer decades later. At each boundary, the operating system, storage technology, computer language, and coding style may change. If this is to work and to be maintainable, any notion of the meaning and representation of 'citizen date of birth' must be independent of the information systems and technologies that persist and present it: we must communicate understanding in a way that transcends technology, purpose, language, and time. These requirements have been clearly presented in *Roadmapping eGovernment Research* [4, §5.3.5, §5.3.19], but are shared in many application domains, notably biomedicine.

While complete solutions to these problems do not exist, much can be done today to make incremental benefits to the interoperability of government services. Today's systems are built vertically within departments, but citizens using these services tend to have horizontal needs: accessing services from multiple agencies [19]. Real vocabulary and data standards efforts in the UK, Australia, New Zealand and US, together with demonstration projects within eGoverment such as SemanticGov [26], TerreGov [28], OntoGov [24], SmartGov [15], and Access eGov [1], and related semantic frameworks for bioscience including the Cancer Common Ontological Representation Environment [18] and the work of the UK CancerGrid project [2] which is indirectly presented here, clearly show ways in which data can be shared between silos, how services can be integrated to better align current systems to citizens needs, and how new systems can be built for interoperability, arguably with less effort than using current development paradigms.

This workshop and tutorial take a pragmatic look at the sorts of technologies, tools, and approaches that could be used today to begin to join up existing silos, and to produce new, well-characterised information systems. In Section 2, we provide an overview of the tutorial, in which we will demonstrate how ISO and W3C standards can support *semantics-driven engineering*, allowing loosely coupled data warehouses that integrate existing information systems. In Section 3, we relate semantics to many of the key themes and 'hot topics' in electronic government. Section 4 presents an overview of the workshop and its contributing papers. The paper is completed by brief conclusions.

2. TUTORIAL OVERVIEW

2.1 Why doesn't it just work?

If we wish our information systems to be accurate and to display some degree of autonomy, then our systems must — in some sense — be able to understand and to reason about our data. To this end, semantic technologies have been used by governments for decades: in the 1970s, an IBM researcher described how one could store any number of facts into a knowledge base, and use these facts to infer new ones through the use of a language that incorporates elements of set theory and logic [5]. Of course, the researcher was Ted Codd, and the system described was the relational database. The core capabilities of the RDBMS remain the same: assertions about an entity identified by a unique key are stored, manipulated, and transformed to infer new assertions that are consistent with the facts that are already known in the database. The strong guarantee of logical consistency has played a large part in the success of the RDBMS, and they have become the key component of information silos.

What has taxed researchers is how we can release information from these databases, and share and reuse the data they contain across the enterprise. Many tools have been developed to extend the use of relational data: ODBC [21] and JDBC [27] allow communication between RDMBSs, middleware and client applications; publication and replication technologies allow the same facts to be maintained in many database instances simultaneously; business intelligence tools such as spreadsheets, online analytical processing, and data mining allow us to access, transform, infer from, and report on data held in a variety of RDBMSs. So why can't we simply connect together all the database instances within a government to achieve 'joined-up Government'?

There are a number of problems: the implementation of Codd's research has never been standard — queries formulated for one RDBMS will not compile and run on another—or complete [7]; axioms about the data are stored in the schema of the database itself — disagree with one of these fundamental axioms and the data is difficult to use in situ; these axioms, and other database metadata, are not exposed in a standard, readily computable form; the query language offers only basic logic and set operations; the theory of tables makes structures that reflect subsumption relationships awkward; problems that have semi-structured solutions — the uniform management of a number of diverse document types, for instance — are difficult to implement; logic that should be implemented in the database is often elsewhere in Java or C# code making it difficult, if not impossible, to understand the entire context of the facts stored.

2.2 Semi-structured data

In parallel to the development of structured databases have been semi-structured, markup-based approaches. Developed in the 1960s, IBM's *Generalised Markup Language* lead to the *Standard Generalised Markup Language* [11] (of which HTML 1.0 is a notable instance), which in turn lead to the widely adopted, simplified *eXtensible Markup Language* (XML) [41], which became a W3C recommendation in 1998.

XML has interesting properties: it is a meta-language—a language for defining languages. When data models are defined in XML — XML Schema and UML models in XMI — they may be communicated along with the raw data to which they conform, and manipulated alongside the data. Thus, data processing may be informed by the data model, in the same way that the axioms encoded into the structure of an RDBMS constrain the results of queries upon the data stored within the table structure.

An important component of the semantics of XML is the namespacing mechanism [42]. Namespaces qualify XML element and attribute names, so that multiple XML vocabularies may be used in a single XML document. In the recommendation, XML element and attribute names may be assigned an *expanded name*, consisting of a namespace name—a URI—followed by a local name that is in the vocabulary specified by the namespace. For brevity, namespace names may be assigned an abbreviation or *namespace prefix*, allowing the use of *qnames* or qualified names in XML documents. Thus the qname '`xsd:element`' abbreviates the expanded name '`http://www.w3.org/2001/XMLSchema | element`'.

Namespaces may be effectively used to partition an eGovernment vocabulary along administrative boundaries: each partition is then free to manage its own schema without central coordination. However, this does not remove all obligation for central coordination of element and attribute names, as identical names in different namespaces are confusing.

Extending the capabilities of namespaces, W3C XML Schema [37] allows one to place constraints upon how an XML vocabulary can be used. Although perfectly applicable to documents with mixed content — annotated text such as an HTML document — XML schema offers more structured data modelling capabilities including inclusion, inheritance, and extension. XML Schemas have been used extensively in e-Government, forming a key component of electronic government interoperability frameworks (e-GIF) in many countries including in the UK [30], New Zealand [31], Australia [29], the US (especially NIEM [33]), and the Danish InfoStructureBase repository [6]. All of these frameworks share the same approach: XML Schema encodes models and model fragments for the validation of XML documents exchanged between systems and departments, and provides an elementary machine-comprehensible representation of data within the domain.

Since XML, XSLT [35, 46], XML Schema, XMI [14], XQuery [45], XForms [44], and WSDL [36] are themselves standard, they offer guarantees of interpretability and the ability to substitute components — for example, the XML Schema specification and compliance tests guarantee conformance of schema validation components, in much the same way that we expect M8$\times$1.25 nuts and bolts to be interchangeable [12]. Tools developed against each of these standards may be reused in a wide variety of projects, and thus can be extremely reliable. Programs specified using such components are simpler than those which implement functionality in a idiosyncratic way, and thus are easier to specify, design, code, test, maintain, and understand. While these properties are desirable in all software development projects, the particular demands for universality, transparency, and security are well served in e-Government applications.

Finally, each of these standards is a computer language that is, or may be, represented in XML. In this way, one may take an object model in XMI and transform it using XSLT into an XML Schema and into an XForm, or take a UML activity diagram into a BPEL workflow specification that may be enacted by any BPEL engine. This reduces the time required to develop applications considerably. If the actual application code is generated in this fashion, then the model of the system is faithfully and automatically transcribed into code, with attendant benefits for transparency, trust, and evolvability.

2.3 Semantic web technologies

While some of the problems associated with the sharing of relational data are avoided by using XML, axioms are still encoded into the structure of the XML document. To overcome this problem, the *Resource Description Framework* (RDF) [34] represents each fact as a triple of subject, predicate, and object, or more informally, the object, property, and value. In this way, all the facts in any database — including those facts encoded in the structure of the database — may be presented in terms of the same simple primitives, and queried together using a standard query language such as the *Simple Protocol and RDF Query Language* (SPARQL) [47]. All database systems have three capabilities: data storage, queries, and consistency. In the most general form, RDF has its own schema language *RDF Schema* [39], which allows one to encode axioms about the class, subsumption, instance, and constraints upon RDF triples.

Further developments have seen subsets of first order logic and set theory more fully expressed in terms of RDF. The *Web Ontology Language* (OWL) [38] allows one to make more detailed models that include well-defined notions of class, subsumption, equivalence, and set membership. OWL models can be tested for consistency, the consequences of assertions explored, and individuals classified automatically. OWL comes in three levels of expressivity: Lite, DL and Full.

A further ontology technology of interest is the *Simple Knowledge Organisation System* (SKOS) [40]. SKOS is an RDF vocabulary designed to represent terminologies, thesauri and classification schemes — its relational semantics are considerably simpler than OWL, admitting only *broader*, *narrower* and *related* assertions, none of which are assumed to be transitive. However, we believe that SKOS has great applicability in e-Government: the problem with highly axiomised OWL models is that they apply to a limited world view, and they can be quite fragile. While we could imagine a law enforcement agency, a public prosecution agency, and defence lawyers sharing a common vocabulary, definitions, and broad associations, detailed models of behaviour would be quite different in each of the circumstances.

Finally, the *Semantic Annotations for WSDL and XML Schema* (SAWSDL) standard [43] provides for the annotation of Schema and WSDL documents with terms from an ontology, and thus provides a framework for the transformation of XML documents into RDF.

2.4 Application to e-Government

How we might use these technologies to achieve 'joined-up government' should now become apparent. For silo computing within a departmental system, traditional solutions may be applied to achieve standard business objectives. Newer, semantic web technologies support semi-structured requirements and intersystem communication where a specific point-to-point information flow is required. Where requirements for integration, querying and inference are more ad hoc, RDF, RDFS, OWL and SPARQL are available. Web technologies may also be used in a generative fashion — if not to generate components of a full solution today, then to provide 'wire-frame' models with little effort to elicit requirements and to test assumptions about an information system under development tomorrow.

2.5 ISO 11179

Nevertheless, these technologies will not be effective unless the meaning of the tags, data items, and models of data are properly described, coordinated, communicated, and reused between designers, developers, and users of the information systems. Another important consideration is that as the same item of data may be held in a spreadsheet, a relational database, an XML database, and in text exports from legacy systems, we need to be able to describe the item of data in an abstract, platform-independent fashion. In this regard alone, requirements supersede those of earlier approaches in the style of e-GIF, where data is only described when in the XML domain. This description must also be computable, in the sense that it may be transformed variously into a series of SQL Data Definition Language (DDL) statements, a UML class diagram, an XML Schema, and a set of OWL classes. Only in this way may one begin to guarantee that the meaning of key measurements, observations, identifiers and attributes are preserved across the enterprise.

We have been supplementing semantic web technologies with a metadata registry based upon the ISO/IEC 11179 standard [13] to overcome these and other limitations. ISO 11179 allows a community to maintain a view of its semantics and metadata independently from the technologies that implement it. In the case of W3C schemas, the documentation and maintenance of rich sets of attribute and element names and data types is considerably simplified: a standard metadata schema for simple types — value domains — and for elements and attributes — data elements — is provided, the relationship of data elements and value domains to models of usage and meaning, classification schemes, and terminologies is made both standard and explicit, and rich administrative workflows are supported. Elements of this approach are incorporated in the US National Information Exchange Model (NIEM) [33]. However, there are distinguishing features between the implementation of NIEM and the approach promoted in this tutorial:

naming and identification ISO 11179 Part 5 describes principles and guidelines underlying naming conventions that may be used in a compliant metadata registry. In an 'informative annex', Part 5 also describes an example naming convention. NIEM implements a naming standard similar to this example: data element names are derived from the names of components in the conceptual half of the ISO 11179-3 metamodel.

The problem with naming conventions is that terminology and usage drift over time and across geographical and administrative distance, even when both are actively and properly managed. Naming rules cannot apply to different languages: noun construction in English is fundamentally different to that in German, so multilingual registries face problems. The naming of data elements from a conceptual model has particular problems in that where the conceptual model needed to change, the names would be rendered nonsensical. An extreme example of the problems this might cause is in the Universal Data Element Framework [32], where the loose conceptual model that frames the Object Classes and Properties clearly requires attention if it is to have a hope of being 'universal'.

In ISO 11179, the globally unique identifier for a data element is the combination of the data element's registration authority identifier, the data identifier — a registry wide unique key — and a version number. We recommend that an e-Government registry simply needs to comply with ISO 11179-6:2005 Annex A (Registration), ensuring that each registry within government has a unique registration authority identifier, each data identifier within a given registry is also unique, and a consistent transformation is applied between its ISO 11179 identifier and a URL.

XML-only implementation The NIEM has a W3C XML Schema-based implementation and tooling; we believe this falls short of what will eventually be required in e-Government. Data management must reach right to the point of data capture — the forms and spreadsheets where users enter data — if it is to ensure data quality sufficient for automatic data integration and reuse. In our tooling, users of artefacts defined using the registry may access definitional content from the registry while entering data to help them. This is expected to reduce requirements for training, and improve the quality of data gathered. By restricting NIEM to schema harmonisation, an important opportunity has been missed. Furthermore, by only describing data that is exchanged, it is not possible for users of the NIEM framework to discover what data actually exists, what may be available on special request, and how other data may have contributed to the derivation of the data exchanged.

data elements and data element concepts ISO 11179-3 describes the qualified assembly of object classes and properties as 'Data Element Concepts'. A data element concept is a notion of a type of data devoid of any representation component — data element concepts have no units, data type, or enumerations. A data element concept may associate a number of actual data elements: one could imagine data for a person's height being recorded in metres, feet, or some value partition of 'short', 'normal', and 'tall'. Each combination of the concept of 'person height' with a concrete representation results in a different data element. If each data element has only one representation, then every data element concept will have only one data element, and thus the data element concept name may also be taken to be a data element name. However, it seems unreasonable — particularly when a government department wishes to record the meaning of data to which it has access but no control — to conflate these two classes.

A suitably designed metadata registry may be used to generate XML schemas. We have been working with members of the UK government to uplift e-GIF into ISO 11179 and to re-generate `simpleType`, `element` and `attribute` libraries — together with appropriate `sawsdl:modelReference` tags within a given namespace from the registry. In this way, the management of the semantics of many components of e-GIF are brought into an administrative domain, where detailed XML Schema skills are not required.

Metadata is also an essential requirement for model-driven software generation. In generative processes, a model is transformed into program code that assembles a set of pre-prepared components into an information system. Generated code often requires customisation: important customisations may be stored in the metadata registry, alongside the semantics of the components of the models. We have been experimenting with the registration of XForm controls in our metadata registry; in this way we can generate a stylesheet that takes raw, generated XForms and applies consistent user interface features.

2.6 Technology walkthrough

The tutorial will consist of a set of linked demonstrations of how semantic web technologies and metadata registries may be used to support data sharing in e-Government. In the demonstration, a simple set of operations will create semantically well-described XML data, which will be automatically transformed into RDF:

- a simple UML model will be designed and annotated with data elements and terms from appropriate e-Government;
- the model will be used to generate an XML schema, using the in-built features of the UML modelling program;

- SAWSDL annotations will be retrieved from the model into the schema, using a simple stylesheet transformation;
- the annotated schema will be transformed into an XForm, which will be customised using another XSLT to enforce stylistic standards upon the form and to target it at a particular application framework;
- the form will then be used to enter data, and a sample dataset will be validated against the schema and then transformed into RDF triples for subsequent data exchange and mining.

Stepping back from the practical demonstration of the kinds of tools we believe are required by e-Government in the future, this approach addresses the explosion of program code required to develop and connect an ever-increasing sea of data and applications, through abstraction of design, code reuse, and coordination of skillsets. Behaviour is modelled more and more in the abstract domain of class models and workflows; generative frameworks target existing, standard software components such as XML validators, XSL transformation engines, and logical reasoners, in addition to the more traditional tools such as RDBMSs; new collaborative tools and cross-discipline interfaces allow system architects and information modellers to work together and reuse each other's work.

If we expect our e-Government systems to be parameterised by — rather than paralyse — policy, then they must be easier to develop, more flexibly configurable, and offer more functionality than is the case today. Each of the approaches described above has already been shown to contain and reduce development cost, reduce errors and reworking, and expose more of the functionality of a system, in a smaller number of documents that may be more easily comprehended.

3. OPEN THEMES

Many of the thirteen key research themes identified by Wimmer and Ma [4, Chapter 6] revolve fundamentally around semantic issue. Some themes directly depend on data semantics: *semantic and cultural interoperability of public services* entails abstract and language-independent representations of the meaning of data, and *ontologies and intelligent information and knowledge management* facilities such as search, retrieval, visualisation, text mining, and intelligent reasoning require common reference models as a foundation. Other themes depend indirectly on semantics: establishing and preserving *information quality* and *assessing the value of government ICT investments* both involve the recording and maintenance of accurate metadata. There are concerns with information system development: *cyber-infrastructures for eGovernment* might consist of libraries of readily available and reusable components conforming to standard data models from which future government applications might be assembled. Questions of social inclusivity encourage agility and flexibility in the design of government services: *crossing borders and the need for governance capabilities* requires the customisation of public administration systems to support multiple perspectives, and *eGovernment in the context of socio-demographic change* requires the streamlined fast adoption and integration of such systems, which in turn depends on clear characterisation of the data they process. Some themes involve matters of openness, which is furthered by being explicit about semantics: *trust in eGovernment* is engendered by transparent models of e-Government processes and services, and *eParticipation, citizen engagement and democratic processes* depends on accountability of public servants. Last, but by no means least, security questions about *government's role in the virtual world* and *data privacy and personal identity* can only be addressed by having clear descriptions of who the users are, what the data means, and who is allowed to access which — true security has to be based on more than syntax.

4. WORKSHOP OVERVIEW

The workshop provides the opportunity to hear about developments in four aspects of semantic technology for electronic government, concerning: the markup of resources with metadata to facilitate information retrieval; the construction and exploitation of an ontology of goals and services to promote the selection of an appropriate service for a particular goal; issues surrounding the maintenance of semantic metadata supporting multiple points of view, whether of multiple organisations or of one organisation at multiple times; and artificial intelligence techniques for analysing interactions with web-based applications so as to distinguish legitimate traffic from threats to security.

4.1 TransEuropean Access to National Case Law: The Caselex Project

Faro and Nannucci [10] describe the Case Law Exchange (*Caselex*) project [3], an information service funded by the European Commission under the *eContent* and *eTen* programmes to provide legal professionals with access to national and European case law without requiring multilingual or multinational legal skills.

Caselex integrates decisions of Supreme and High Courts of Member States from disparate national sources into a single combined multinational resource; it focusses particularly on commercial law. The full text of each case is provided in the native language of the case, together with an additional headnote, summary, and metadata in English. Caselex exploits the thesaurus of terms relevant to the legal area it covers, and cases are annotated with terms from this thesaurus; users of Caselex may use these terms to identify cases of interest.

This paper demonstrates the use of semantic metadata in electronic government.

4.2 Goal-Oriented Service Selection

Salhofer, Tretter and Stadlhofer [25] present an approach to allowing digital citizens to express a goal for interaction with government, and to assisting them in selecting the appropriate electronic government service to achieve this goal. They phrase this exercise in terms of the 'Government Enterprise Architecture — Public Administration' (*GEA-PA*) metamodel for public services.

The approach is based on capturing the ontology of the domain, for which they use the Web Service Modeling Ontology (*WSMO*). The goal discovery process starts with a goal template corresponding to a root concept in this ontology, and then directs the user in refining template until an appropriate service can be unambiguously identified. The initial prototype is limited to the domain of obtaining approval for a construction project, and the presentation is illustrated with the example of discovering the appropriate form of approval for such a project.

This paper provides an application to electronic government of higher-order semantic modelling, in the form of an ontology.

4.3 Metadata Standards for Semantic Interoperability in Electronic Government

Davies et al. [9] consider how the ISO 11179 standard for metadata registration can be employed in electronic governance, without the need for prior universal agreement on a common model of governance. They identify a number of problems that may arise, particularly concerning semantics from multiple perspectives or under changing usage. They also identify solutions to those problems, by extension and specific interpretation of the ISO 11179 standard.

The result is an approach to metadata standards that supports multiple perspectives, an essential prerequisite for semantic interoperability across separate participants; the value of this approach with respect to initiatives such as the UK e-Government Interoperability Framework (*e-GIF*) is discussed.

This paper addresses the issues in lifting a semantic technology intended for a single perspective to work for multiple perspectives.

4.4 Proposing a Hybrid-Intelligent Framework to Secure E-Government Web Applications

Moosa and Alsaffar [22] discuss approaches to securing web applications, such as those implementing electronic government applications. They propose a hybrid intelligent web application firewall (*HiWAF*) approach, combining several successful concepts underlying web application security. Their proposal is to use ideas from artificial intelligence in web traffic filtering: they suggest using artificial neural networks to implement 'negative logic filtering' for rejecting traffic that matches known attack profiles, and ideas from fuzzy logic to implement 'positive logic filtering', allowing only traffic that conforms to accepted patterns of usage.

Security is a question of semantics too; this paper is concerned with the analysis of web traffic to determine whether it is legitimate or not.

5. CONCLUSIONS

We believe that semantic technologies are fundamental to successful electronic government. This tutorial and workshop explores that hypothesis: the tutorial presents and in-depth study of a variety of semantic technologies, and illustrates their application in electronic government scenarios; the workshop consists of four papers reporting on specific questions in the application of semantic technologies.

6. ACKNOWLEDGMENTS

We wish to thank all authors of the workshop papers for their contributions.

7. REFERENCES

[1] Access-eGov. Access to e-government services employing semantic technologies. http://www.accessegov.org/.

[2] James Brenton, Carlos Caldas, Jim Davies, Steve Harris, and Peter Maccallum. CancerGrid: Developing open standards for clinical cancer informatics. In *Proceedings of the UK e-Science All Hands Meeting 2005*, pages 678–681, 2005. http://www.allhands.org.uk/2005/proceedings/.

[3] Caselex service. www.caselex.org.

[4] Cristiano Codagnone and Maria A. Wimmer, editors. *Roadmapping eGovernment Research: Visions and Measures towards Innovative Governments in 2020*. Guerinoni Marco, 2007. ISBN 978-88-95549-00-2.

[5] Edgar F. Codd. A relational model of data for large shared data banks. *Communications of the ACM*, 13(6):377–387, 1970. http://doi.acm.org/10.1145/362384.362685.

[6] Danish Government. Infostructure base repository. http://isb.oio.dk/repository/.

[7] Chris Date. *An Introduction to Database Systems, 8th ed.* Addison Wesley, 2004. ISBN 0-321-18956-6.

[8] Jim Davies, Jeremy Gibbons, and Steve Harris, editors. *Workshop on Technology in Electronic Government at ICEGOV2008*, Cairo, December 2008.

[9] Jim Davies, Steve Harris, Charles Crichton, Aadya Shukla, and Jeremy Gibbons. Metadata standards for semantic interoperability in electronic government. In Davies et al. [8].

[10] Sebastiano Faro and Roberta Nannucci. TransEuropean access to national case law: The Caselex project. In Davies et al. [8].

[11] International Organization for Standardization. Information processing – text and office systems – standard generalized markup language. ISO/IEC 8879, 1986.

[12] International Organization for Standardization. General purpose metric screw threads – general plan. ISO/IEC 261, 1998.

[13] International Organization for Standardization. Information technology – metadata registries (MDR). ISO/IEC 11179, 1999. http://metadata-standards.org/11179/.

[14] International Organization for Standardization. Information technology – XML metadata interchange (XMI). ISO/IEC 19503, 2005.

[15] Panagiotis Georgiadis, Giorgos Lepouras, Costas Vassilakis, G. Boukis, Efthimios Tambouris, Stelios Gorilas, Elisabeth Davenport, Ann Macintosh, John Fraser, and Dave Lochhead. SmartGov: A knowledge-based platform for transactional electronic services. In *EGOV'02: Proceedings of the First International Conference on Electronic Government*, pages 362–369, London, UK, 2002. Springer-Verlag.

[16] Great Britain Cabinet Office. *Transformational Government: Enabled by Technology*. The Stationery Office, November 2005. ISBN 9780101668323. http://www.cio.gov.uk/transformational_government/strategy/.

[17] J. S. Kilby. Miniaturized electronic circuits. US Patent Number 3,138,743, United States Patent and Trademark Office, June 23 1964. http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=3138743.

[18] George A. Komatsoulis, Denise B. Warzel, Francis W. Hartel, Krishnakant Shanbhag, Ram Chilukuri, Gilberto Fragoso, Sherri de Coronado, Dianne M. Reeves, Jillaine B. Hadfield, Christophe Ludet, and Peter A. Covitz. caCORE version 3: Implementation of a model driven, service-oriented architecture for semantic interoperability. *Journal of Biomedical Informatics*, 41(1):106–123, 2008. http://www.sciencedirect.com/science/article/B6WHD-4ND60KV-1/2/b1ab108b1b63c80298de8fbe930622c8.

[19] Lorenzo Madrid. The economic impact of interoperability. Technical report, Microsoft Corporation, January 2008.

[20] Andrea Di Maio. Move 'Joined-Up Government' from theory to reality. Technical Report G00123844, Gartner Research, October 2004. http://www.gartner.com/DisplayDocument?doc_cd=123844.

[21] Microsoft. Open Database Connectivity (ODBC) overview. http://support.microsoft.com/kb/110093.

[22] Asaad Moosa and Eanas Muhsen Alsaffar. Proposing a hybrid-intelligent framework to secure e-government web

applications. In Davies et al. [8].
[23] R. N. Noyce. Semiconductor device-and-lead structure. US Patent Number 2,981,877, United States Patent and Trademark Office, April 25 1961. http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=2981877.
[24] Ontology-enabled e-Gov service configuration. http://www.ontogov.com/.
[25] Peter Salhofer, Gerald Tretter, and Bernd Stadlhofer. Goal-oriented service selection. In Davies et al. [8].
[26] SemanticGov. Providing integrated public services to citizens at the national and pan-European level with the use of emerging semantic web technologies. http://www.semantic-gov.org/.
[27] Java Database Cconnectivity (JDBC) overview. http://java.sun.com/products/jdbc/overview.html.
[28] TerreGov. Impact of eGovernment on territorial government service. http://www.terregov.eupm.net/.
[29] The Australian Government. Technical interoperability framework. http://www.finance.gov.au/e-government/service-improvement-and-delivery/technical-interoperability-framework.html.
[30] The Cabinet Office. e-Government interoperability framework (e-GIF). http://www.govtalk.gov.uk/schemasstandards/egif_document.asp?docnum=949.
[31] The New Zealand Government. New Zealand E-Government Interoperability Framework. http://www.e.govt.nz/standards/e-gif.
[32] The Open Group. Universal data element framework. http://www.opengroup.org/udefinfo/obj.htm.
[33] US Government. National information exchange model. http://www.niem.gov/.
[34] W3C. Resource description framework (RDF) model and syntax specification. http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/, 1999.
[35] W3C. XSL transformations (XSLT), version 1.0. http://www.w3.org/TR/xslt, 1999.
[36] W3C. Web services description language (WSDL) 1.1. http://www.w3.org/TR/wsdl/, 2001.
[37] W3C. XML schema. http://www.w3.org/XML/Schema, 2001.
[38] W3C. OWL web ontology language. http://www.w3.org/TR/owl-features/, 2004.
[39] W3C. RDF schema. http://www.w3.org/TR/rdf-schema/, 2004.
[40] W3C. SKOS simple knowledge organization system. http://www.w3.org/2004/02/skos/, 2004.
[41] W3C. Extensible markup language (XML) 1.0 (fourth edition). http://www.w3.org/TR/xml/, August 2006.
[42] W3C. Namespaces in XML 1.0 (second edition). http://www.w3.org/TR/REC-xml-names/, August 2006.
[43] W3C. SAWSDL semantic annotations for web service definition language. http://www.w3.org/2002/ws/sawsdl/, 2007.
[44] W3C. XForms 1.0, third edition. http://www.w3.org/TR/xforms/, 2007.
[45] W3C. XQuery 1.0: An XML query language. http://www.w3.org/TR/xquery, 2007.
[46] W3C. XSL transformations (XSLT), version 2.0. http://www.w3.org/TR/xslt, 2007.
[47] W3C. SPARQL query language for RDF. http://www.w3.org/TR/rdf-sparql-query/, 2008.