Technological Foundations of Electronic Governance

Jim Davies
University of Oxford
Wolfson Building, Parks Road,
Oxford OX1 3QD UK
+441865283521
jdavies@comlab.ox.ac.uk

Tomasz Janowski, Adegboyega Ojo
Center for Electronic Governance
United Nations University IIST
P. O. Box 3058, Macao
+853 28712930
{tj,ao}@iist.unu.edu

Aadya Shukla
University of Oxford
Wolfson Building, Parks Road,
Oxford OX1 3QD UK
+441865283521
aadya@comlab.ox.ac.uk

ABSTRACT
This paper explores the relevance and opportunities for the application of mature Formal Techniques – techniques based on mathematical theories and supported by industry-ready tools and methods – to build technical solutions for Electronic Governance. The paper proceeds in four steps: (1) establishes the basic need for Formal Techniques in Electronic Governance, (2) identifies the challenges peculiar to Electronic Governance development, (3) presents the salient features and various application scenarios for Formal Techniques in general, and (4) carries out a mapping between the challenges to Electronic Governance and various application scenarios of Formal Techniques as part of solutions to such challenges. In the second part, the paper presents an overview of the tutorial and workshop on Formal Engineering Methods for Electronic Governance. The tutorial follows the four-step program, as above, and the workshop includes the presentations of four papers that exemplify various elements of the mapping, particularly: the use of formal, precise modeling techniques; the importance of security risk assessment; model-driven development of software systems; and the provision of semantic frameworks to coordinate development within and across major programs and initiatives. In the last part, the paper discusses how Formal Techniques can contribute to establishing a solid foundation for Electronic Governance.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**], D.2.1 Requirements/ Specifications, D.2.12 Interoperability

General Terms
Design, Documentation, Security, Standardization, Theory

Keywords
Electronic Governance, Formal Techniques

1. INTRODUCTION
The focus for Electronic Government research and practice has been shifting over the years. Initially, the centre of attention was overcoming technical challenges to Electronic Government implementation, inline with the first wave of public websites and services going online. As soon as the basic level of online presence was assured and governments began implementing more mature, transactional and seamless services, the extent of organizational challenges to be addressed quickly surpassed purely technical issues, especially in view of traditional government structures and modes of operation. This plus the growing realization that IT investment has to be measured in terms of creating public value caused the focus to shift towards organizational issues, including the alignment of Electronic Government initiatives with the broad Public Sector Reform agenda. However, the availability of mature, cross-agency public services does not mean that such services will be automatically used and therefore how much public value has been actually created. Thus the focus has shifted again, now towards social issues, and the scope expanded from Electronic Government – technology-enabled improvements in government operations, to Electronic Governance – technology-enabled improvements in interactions between government and non-government actors.

The changing focus and priorities for Electronic Governance research and practice, coupled with the low maturity of the area underscores in our opinion the importance of explicit efforts aimed at building foundations for Electronic Governance, particularly through establishing formal connections with more mature areas of science and technology. The tutorial-workshop event on Technological Foundations of Electronic Governance highlighted by this paper contributes to this aim, focusing particularly on technical solutions. The event explores the relevance and opportunities for the application of mature Formal Techniques – techniques based on mathematical theories and supported by industry-ready tools and methods – to build technical solutions for Electronic Governance.

The tutorial proceeds in four logical steps to realize this goal. The first step is to establish the basic need for applying Formal Techniques in developing technical solutions for Electronic Governance, appealing to the classical way such techniques are applied in software processes: specify desirable functionality, build a model of the implementation, and verify that the implementation satisfies the specification. The second step explores a range of peculiar technical challenges presented by Electronic Governance, generally absent or less prominent in other kinds of applications. An example is strict dependence of electronic public services on changing law and regulations that govern how such services must be delivered. Another example is extreme heterogeneity of government systems, due to the long-term, mission-critical nature of many public services, accumulation of technical solutions to deliver them, and

ICEGOV2007, December 10-13, 2007, Macao

participation of many agencies from diverse functional areas of the government. The third step is to present the salient features and various application scenarios for Formal Techniques in general, beyond the classical specify-design-verify paradigm put forward in step (1). An example is model-driven development - making possible for software to be automatically generated from abstract models, and regenerated every time such models change. Another example is building ontologies to formalize the concepts exchanged by different systems, so that such concepts, albeit described with different syntaxes, are understood by different actors in the same way. The fourth step is to carry out a mapping of the challenges to Electronic Governance identified in the step (2) to the application scenarios of Formal Techniques produced as part of step (3). An example is the application of model-driven development to address the dependence of electronic public services on changing law and regulations, relying on the model to formalize such dependencies. With every change of the law, the model will be updated and software regenerated from the model. Another example is using ontologies and ontology-mappings to ensure that highly heterogeneous systems run by government agencies are semantically interoperable, despite the agencies' use of own specialized terminologies and representation languages.

The workshop, in contrast, builds upon the framework and the body of concepts established through the tutorial, focusing on the step (4). It offers a selection of four papers that exemplify various scenarios for the application of Formal Techniques to Electronic Governance. The tutorial if targeted at new-comers to the area, particularly at government technology researchers, educators and solution developers. The workshop is particularly for researchers interested in state-of-the-art, but also aims at network-building.

The rest of this paper is organized as follows. Sections 2 through 5 correspond to the four steps described above. In particular, Section 2 establishes the relevance of Formal Techniques to Electronic Governance, Section 3 explores technical challenges presented by Electronic Governance, Section 4 presents novel application scenarios for Formal Techniques, and Section 5 carries out a mapping between the challenges and application scenarios. The final Section 6 provides an overview of the event.

2. RELEVANCE

In one view, technological challenges of Electronic Governance are no different from those found in large-scale computerisation projects within the private sector. If so, it would seem sensible, then, to concentrate upon developing our understanding of the problem domain, to better manage the introduction of existing, appropriate, commercial solutions.

However, the requirements of Electronic Governance extend strictly beyond those of the business world, in terms of flexibility, transparency, and interoperability, and the appropriate commercial solutions do not exist. The demands of commerce will not produce the technology required; instead, government must drive its development – as an intelligent customer, and as an active partner in the design of e-Government solutions.

To understand precisely why the requirements for Electronic Governance should be in advance of those for the computerisation of activities within large multinational corporations and industry consortia, it is useful to consider their source: they reflect the collective needs and ambitions of our society, expressed through a combination of legislation and public opinion. The extent of the collaboration required is far greater, and the outcomes are much further removed from the immediate business objectives of the participating organisations.

The need for active engagement, on the part of government, in the design and development of solutions stems from the lack of any clear incentive for commercial software providers to close the technology gap. Interoperability with solutions delivered by a rival company is often a mixed blessing, as is flexibility; it is one thing to be able to update your software, quickly and easily, and sell a new release to a customer; it is quite another to see the customer updating it themselves, or finding that it can be easily reconfigured to meet a new formulation of requirements.

To drive the development of the technology, it is essential that governments, or their agents, be able to produce precise specifications of functionality. This kind of specification requires an approach to modeling known as *formal engineering techniques*: the use of mathematically-based languages, methods, and tools. Informal, imprecise use of graphical notations such as flow diagrams is not enough: the meaning of the specification must be clear and irrefutable.

Naturally, there may be no great advantage to presenting a precise specification that turns out to be inconsistent with the actual requirements. To do so could remove all authority from the specification, and put the supplier in a powerful position, able to apply significantly increased costs to a contract that they have already been awarded. The specification itself must be tested, and the consequences of high-level design decisions explored, in advance of procurement or development.

The need for formal meaning in the specification extends beyond the interpretation of the modelling language, and thus the relationships between the terms used, to include an informal – but agreed – understanding of the terminology and a structured, computable description of the context in which the measurements and observations are made. This kind of meaning is provided by way of a semantic framework, consisting of definitions and models created and maintained by a community of interest.

3. ELECTRONIC GOVERNANCE

3.1 Concepts

Government as an institution of the state is responsible for: (i) provision of common good - public services and infrastructure, (ii) governance of the state – rule making, implementation and adjudication and (iii) maintenance of social order and security [1]. By rules, we mean the laws, policies, regulations and programs of the government. The public administration system is responsible for the delivery of public services and provision of public infrastructure, while implementing the rules under direction of the executive branch of government. Rules are made and adjudicated by the legislative and judiciary organs of government respectively.

For conceptual simplicity, we shall assume that the main function of government is governance and consequently that the notion of governance covers all processes for public policy development (rule making) and public service provision (including public infrastructures). We further assume that the maintenance of social order and security are some kinds of public services (control services). There are various categories of policies including -

social, economic, environmental and cultural policies [1][2]. Public services may also be organized into four categories [2]: Certification (e.g. birth and marriage registry services), Control (e.g. inspection or security services), Authorization (e.g. licensing or welfare services) and Production (e.g. utility services).

The use of information and communication technology has a long tradition in the public sector – spanning over four decades particularly in routine back-office operations [4]. In the last five years, ICT is being deployed to fundamentally transform the relationship between government and the society (equally addressing front- and mid-office operations). We refer to such strategic and transformational application of ICT in governance processes in general as Electronic Governance (e-Governance). In this paper, the term e-Governance subsumes related concepts like e-Government, e-Participation, e-Voting or e-Democracy.

3.2 Intrinsic Features

A well known fact about e-governance initiatives is high failure rate both in terms of development and usage. Several reasons have been adduced for this high rate of failures including the hitherto limited view of e-Governance initiatives as primarily technology-focused. Other reasons are neglect of stakeholders' expectations and neglect of the nature of public management and governance [3]. The latter is explained in the popular expression by Wallace Sayre [4]: "public and private management are fundamentally alike in all unimportant respects". Therefore, addressing this misconception lies in paying specific attention to the differences that exists in the "important aspects". Three of these aspects are identified in [4]: specificity of government tasks, the role of law (normative aspect) and the special significance of knowledge.

The rest of this section discusses in more detail these and others features of public management and e-governance, distinguishing it from private management and e-business. These features are organized into two categories: regulatory and policy aspects, and organizational and technical aspects. While some of these features provide specific challenges which have been overlooked, other features provide opportunities that could be harnessed.

Regulatory and Policy Perspective: A major aspect of public sector governance is the legal structuring of administrative tasks. These laws are normative in nature, communicating the goal to be achieved and specifying the degree of freedom in related decision making. The following six features collectively define the basic challenges and dilemmas peculiar to public governance:

R1) Privacy Protection – Government agencies typically need to collect personal information from citizens when they are applying for public services, particularly through electronic services [5]. Information that has been collected by one agency is generally not expected to be requested by some other agencies in the same government, particularly in the context of e-services and so-called one stop services. However, in the same environment there are very stringent privacy acts protecting citizen's privacy rites through data protection acts – forbidding sharing of citizen information without expressed consent from them. To guarantee non-violation of privacy rights, designated agencies carry out privacy audits on administrative procedures.

R2) Eligibility Criteria – Unlike in the private sector, most public services have stipulated eligibility criteria and requirements. For instance, to provide a social welfare service, the residency status of the applicant must be determined and a number of other documentary evidences supplied [6].

R3) Identity Management – To receive public services, positive identification of applicants is required. Identity verification is key support feature for establishing public service eligibility.

R4) Anonymity Protection – While there is a need for positive identification of citizens in most service interactions, at the same time, there are a number of governance processes such as consultation and voting processes which also demand anonymity. So while a participant must first be positively identified as a citizen to be eligible to vote in a referendum, ballot information must not be traceable or linked to voter's identity. While anonymity is an important feature in private sector and e-commerce, the legal consequences of anonymity violations in the public governance are more severe. In fact, auditability is demanded as part of a voting process [7].

R5) Accessibility Support – There is growing awareness on the need to explicitly support the accessibility of government services and contents to all segments of the society. While in the private sector, a segment of the society (e.g. disabled) may not be targeted, public organizations are required to provide equal access to all. Presently, over 20 countries have developed guidelines for web accessibility (http://www.w3.org/WAI/Policy). In addition, governments of the UK and Australia are deploying other assistive technologies. The failure to comply with accessibility regulations may results in significant legal liabilities in some countries.

R6) Standards Enforcement – Considering the heterogeneity of agency environment in terms of process, technology and management, a few core standards are mandated by governments for streamlining government operations and services. A good example is the accessibility standard (R5). Other standards typically supported by regulations include the provision of IT architecture inline with general guidelines (e.g. in the US Clinger-Cohen Act) and information security.

Perhaps the most challenging aspect of these features is the need for systems supporting governance processes to be able to cope with the highly dynamic nature of the regulatory environment. For instance, public management information systems or electronic public services at any point in time are expected to support the most current regulations or policies related to privacy and anonymity, eligibility, identification and accessibility.

Organizational Perspective: These features identify the peculiar requirements of governance processes in terms of information sharing and collaboration, the nature of IT function in government agencies, types of services delivered by the core business units of government agencies and the nature of the specialised knowledge required for decision making in administrative procedures:

O1) Collaboration – Government agencies need to engage in several forms of collaborations within and outside government. Three forms of such interactions can be easily identified: agency-to-agency, agency-to-supplier and agency-to-intermediary. Collaboration between agencies may involve: (i) provisioning of information - data sharing, (ii) provisioning of technical opinions to support decisions

on applications, (iii) certification of information supplied as evidences and (iv) simple notification of another agency to ensure consistency of data across the public administration. Interactions between an agency and a supplier could also involve the provision of information or technical opinions as well as receiving concrete services from the supplier towards fulfilling administrative goals. Intermediaries act on behalf of agencies by delivering public services and related value-added services. A major challenge in collaborative government processes is how to ensure non-violation of privacy regulations, particularly considering the possibility of citizen's data being in the custody of private sector organizations. In the context of the one-stop services, collaborations often raise other challenges - for instance, responsibility sharing in the event of contract violation [8].

O2) IT Function – Traditionally, IT units in public organizations provide basic support services to IT users in other business units, with limited staff and budget and significant legacies. Within the constraints of limited resources, a typical IT unit of a government agency is presently faced with pressures for larger IT portfolio to respond to the needs for developing new electronic public services and to support organizational transformation efforts. In addition, most IT units are deeply embedded in the organizational structure of agencies with minimal management leverage for the IT Head.

O3) Administrative Services: Typically, a large number of public services is delivered by governments. However, these services are often very similar in terms of the underlying administrative processes. This is clearly evident from the detailed documentation of over 100 public services in [10] and various public administration modelling efforts [2][4]. In [11][12], generic processes resulting from the analysis of 25 licensing and 6 social welfare services were presented. The regularity of major processes underpinning public services is a valuable feature of the domain.

O4) Knowledge – Many of the services offered by governments involve complex decision-making, requiring specialised knowledge. Administrative decision-making draws on the following kinds of knowledge at an operational levels [4]: (i) legal knowledge, (ii) knowledge of the facts given in a special case to be decided upon, (iii) knowledge about the means for action, which government has at its disposal, (iv) knowledge about the effectiveness if various measures and (v) process memory which gradually builds up while working on decision cases. These give rise to knowledge management requirements different from the private sector - knowledge for administrative action in the public domain versus knowledge for innovation in the private domain.

3.3 Technical Implications

Section 3.2 presented regulatory and organizational features of the domain. Here we attempt to draw technical requirements for these features to inform technological solutions. Preliminary analysis reveals seven technical requirements: (i) interoperability of government IT ecosystems, (ii) rapid development of electronic public services (EPS) and public management information systems (PMIS), (iii) Adaptability or evolvability of EPS/PMIS, (iv) rigorous contract specification, (v) specification of security properties, (vi) verifiability of conformance to technical standards, (vii) verifiability of correct implementation of security properties, and (viii) knowledge-based software development. These requirements are briefly explained below.

T1) Interoperability of government IT ecosystem – To ensure effective collaboration (feature O1) within government and between government and third-party entities, interoperability within the cooperation space at the technical, semantic and organizational levels must be guaranteed.

T2) Rapid EPS/PMIS development – The continuous pressure on the IT function to enable the delivery of public services over electronic channels and build new back-office applications within tight time-frames necessitates a rapid development methodology for building electronic services and PMIS (O2).

T3) Adaptability of EPS/PMIS - The dynamic nature of the legal environment for e-Governance and public service delivery in particular (R1–R5) necessitates that the overhead for the modification of EPS/PMIS to align with regulatory changes be minimized. The development method underpinning the EPS/PMIS should prescribe an architecture in which policy components are separated from the application logic.

T4) Rigorous contract specification - The slim nature and limited capabilities of the IT function makes outsourcing of projects imperative (O2). Writing system requirements in outsourcing contracts require clarity and verifiability.

T5) Specification of security properties - Several important and regulatory provisions related to privacy (R1, R4) necessitates a domain specific framework for specifying these properties. The availability of a suitable framework where domain specific requirements for privacy, anonymity and possibly accessibility can be effectively expressed, is a foundation for correct verification of these properties at future time.

T6) Verifiability of security properties – Major security properties of the domain (e.g. R1 and R4) have to be verified to guarantee that the design or implementation of EPS/PMIS do not violate any specified security properties.

T7) Verifiability of conformance to standards – Verification of interoperability (in heterogeneous cooperation spaces – O1), as well as other standards stipulated by regulations is important to guarantee conformance (R5, R6). Providing semi-automated means to carry out verification is desirable since manual verification or audit would be too tedious.

Table 1: Domain Challenges

Regulatory Elements	Organizational Elements
R1) Privacy Protection	O1) Collaboration
R2) Eligibility Criteria	O2) IT Function
R3) Identity Management	O3) Administrative Services
R4) Anonymity Protection	O4) Knowledge Actions
R5) Accessibility Support	
R6) Standards Enforcement	
Technical Implications	
T1) Interoperability of government IT Ecosystem	
T2) Rapid EPS/PMIS development	
T3) Adaptability of EPS/PMIS	
T4) Rigorous contract specification	

T5) Specification of security properties
T6) Verifying security properties
T7) Verifying conformance to standards
T8) Knowledge management

T8) Knowledge-Management – The knowledge intensive nature of administrative services (O4) necessitates an effective mechanism for externalizing, managing requisite knowledge and integrating the knowledge base with EPS/PMIS.

Table 1 above summarizes the features of the e-governance domain and the associated technical challenges.

4. FORMAL TECHNIQUES

A formal technique is a notation with a mathematical semantics: a precise explanation of how it should be interpreted, given in terms of an already-understood area of mathematics – such as algebra, logic, or set theory. This notation is typically accompanied by a method, an account of how it is to be applied to the description or the analysis of computing systems, and will often be supported in this application by one or more tools – model checkers, type checkers, or theorem provers.

For convenience, we extend the definition of formal techniques to cover semantic and knowledge-based techniques. Semantic techniques allows for explicit representation of meanings of and associations between information to be processed at run-time [20]. As we shall show shortly, semantic techniques offer complimentary approaches to formal techniques that are useful for addressing a variety of solutions peculiar to e-Governance. In the rest of this section, we present several scenarios demonstrating the use of formal and semantic techniques. The first six relate to formal techniques and the latter five to semantic techniques.

Formal Techniques Scenarios: The following six scenarios relate to the use of formal techniques to build clear understanding of domains to generate implementations from abstract specifications.

S1) Formal Domain description - We may use formal techniques to build an understanding of the key concepts and intentions independently of the construction or implementation of any technological solution. The result can be an improved description of requirements, or a precise account of an intended design – in either case, we can use methods and tools to extend our understanding, calculating precise consequences of adopting a particular formulation of requirements, or a making particular design decisions.

S2) Formal systems specification – Formal techniques may be used during construction and maintenance of a solution to provide formal models of requirements and design.

S3) Specification-based testing - We may use a formal model of requirements or design as the basis for the construction of a suite of tests, which can then be used to check that an implementation will behave in a manner that is consistent with the agreed specification. In the simplest case, the formal specification may consist simply of precise descriptions of tests, as examples of required behavior.

S4) Behavioral specification - If we wish to be more ambitious, then we may use formal engineering techniques to construct a complete account of intended behavior. A process model – such as that presented by a state machine in UML [14], or a collection of algebraic processes in CSP [15] – may be used to specify all allowable sequences of interactions of a system or its component at a particular level of abstraction.

S5) Formal verification - If we give a formal interpretation also to implementation artifacts, such as service descriptions, program annotations, or programming language statements, we may then be able to prove that every behavior of the implementation is guaranteed to be consistent with the formal specification. The prospect of this kind of proof, or formal verification, provided much of the initial motivation for the development of these techniques, particularly within the domain of critical systems development. However, the process of verification is one that is, in general, impossible to automate: the number of possibilities to be explored, in any but the most trivial of proof or development steps is such that a considerable degree of human interaction, and creativity, is required. This is difficult, expensive, and time-consuming, which can render the whole exercise redundant, as the requirements may change, leaving the specification out of date before we can show that it has been correctly implemented. One solution is to ensure that the processes of development and verification proceed in tandem. If we ensure that our software artifacts are constructed with subsequent verification in mind, and interwoven with descriptions of intended functionality, then we may be able to establish, automatically, that certain properties are satisfied. An example is the use of program annotations in languages such as Spec# [17], and the subsequent application of tools such as Boogie [18].

S6) Generating implementation from formal specifications - Within specific domains, we can go further, and generate the implementation directly from the formal specification: provided that the code generation technology has itself been verified – a simpler task, and one that has to be completed only once for each class of systems – the consistency of specification and implementation is automatically guaranteed. The feasibility of this approach relies upon our ability to represent the process of development as a set of transformations, sufficient for the automatic elaboration of an abstract design into a concrete implementation. [19] shows that this is possible for information systems whose behavior can be characterized in terms of the intended effect and availability of operations, each interpreted as a single transaction upon the system. Pre- and post-conditions in the specification are elaborated, and translated into functions within a programming interface. The approach, and the supporting tool technology, has been used in the construction of several working systems, all of which have a critical business role within an organization.

Semantic Technique Scenarios: These scenarios demonstrate the use of ontologies and knowledge management in domain descriptions, formal annotation of services and components and codification of business rules/policies for automatic verification.

S7) Ontology as a domain object model – In addition to the use of formal techniques for unambiguously capturing domains, ontologies can also be used to establish a shared understanding of any domain. An ontology-based domain object model can be linked to codes to enable dynamic use by components and applications [16].

S8) Ontology based requirement specification - While formal specifications of requirements allows for design verification and generation of test cases, ontology-based specifications offers both semi-formal and formal models of requirements with shared understanding of the requirements. While customers may not understand formal languages, ontologies are generally more accessible. End-to-end use of ontologies in analysis, design and implementation is highly suitable for rapid application development [16].

S9) Semantic components description – Component-based development involves identification of suitable components that could be customized to implement desired functionality. Providing semantic annotations for components and services in registries (through ontologies) enables greater precision in matching functionality needs against services/components.

S10) Ontology as policies, regulations and business rules: The separation of policies and rules from application processing logic enables run-time adaptation of an application to changing environmental contexts. The use of declarative-style rules in the form of ontologies through some knowledge layer or specialized middleware provides important adaptation feature for applications. By formalizing policies and regulations, it is possible to automatically evaluate the relative consistency of policies and regulations [13].

5. FORMAL TECHNIQUES FOR ELECTRONIC GOVERNANCE

We attempt in this section to map the different application scenarios highlighted in Section 4 to the challenges identified in Section 3. Summaries of these mappings are given in Tables 2 and 3. Table 2 maps the scenarios to domain challenges while Table 3 maps the scenarios to technical challenges. Each of the mapping elements is described below:

Scenario S1 – Interoperability between collaboration parties can be enhanced through formal description of entities (e.g. messages), events and actions involved in the interaction context.

Scenario S2 – Contract specifications in outsourced development projects by the IT function can benefit from clear formal specification of the desired system, particularly when specifying privacy and anonymity properties.

Scenario S3 – Verification of sources delivered by third-parties or suppliers can be carried out using test cases generated from formal requirements specifications. In particular, privacy and anonymity properties can be effectively tested if that Scenario S2 is given.

Table 2: Mapping Scenarios to Domain Challenges

No	Scenario	Challenge
S1	Formal Domain description	O1
S2	Formal systems specification	O2, R1, R4
S3	Specification-based testing	O2, R1, R4
S4	Behavioral Specification	O2, R1, R4
S5	Formal verification	R1, R4
S6	Generating from formal specifications	O2
S7	Ontology as a domain object model	O1
S8	Ontology based specification	R1, R4
S9	Semantic components description	O2
S10	Ontology as Policies and business rules	R1-R5, O4

Scenario S4 – Detailed behavioral specification can be provided as part of system description in Scenario S2.

Scenario S5 – Given Scenarios S2 and S4, formal verification can be carried out to ensure privacy and anonymity properties are not violated in the supplied implementation.

Scenario S6 – By generating implementations from formal design models, EPS/PMIS can be rapidly developed.

Scenario S7 – Ontologies provide explicit specifications of various concepts in the collaboration space to enable semantic interoperability between parties.

Scenario S8 – Using ontologies to describe privacy and anonymity requirements enables shared understanding of the requirements and the possibility for system to evolve with changes in privacy and anonymity regulations.

Scenario S9 – development time for EPS/PMIS can be reduced if components and services fulfilling specific functionalities can be easily found in repositories. Semantic annotation potentially enables accurate discovery of components and services.

Scenario S10 – Externalizing policies, regulations and administrative task knowledge as ontologies insulates the EPS/PMIS from effect of changes in the environment.

Table 3: Mapping Scenarios to Technical Challenges

No	Scenario	Challenge
S1	Formal Domain description	T1
S2	Formal systems specification	T4, T5
S3	Specification-based testing	T4, T6
S4	Behavioral Specification	T4, T5
S5	Formal verification	T6
S6	Generating from formal specifications	T2
S7	Ontology as a domain object model	T1,
S8	Ontology based requirement specification	T5
S9	Semantic components description	T2
S10	Ontology as Policies and business rules	T3, T7, T8

6. TUTORIAL-WORKSHOP OVERVIEW

The tutorial aims to introduce the audience to formal and semantic techniques and their applications to the domain of Electronic Governance. It is organized into three parts. The first part will discuss peculiar features of the Electronic Governance domain – from policy through organizational to technical perspectives (Section 3). The second part will survey available

formal and semantic techniques (Section 4) and show their applications. The last part will discuss how the various application scenarios of formal and semantic techniques can be applied to the identified domain challenges (Section 5).

The workshop features four papers that provide concrete examples of how Formal Techniques are able to address specific challenges in Electronic Governance domain.

The first paper on "Formal Support for e-Government System Design with Transparency Consideration" by Xiaoyi Chen, Weiqiang Kong, Kokichi Futatsugi, explores abstractions and concepts such as transparency independently of any existing technological solution. The paper provides better understanding of administrative transparency through formal models.

The second paper "Formal Threat Descriptions for Enhancing Governmental Risk Assessment" by Andreas Ekelhart, Stefan Fenz and Edgar Weippl shows how an ontology for threat definition – a formal model of threats, vulnerabilities, controls, and safeguards – can enable threat identification and response, including automatic updating and reconfiguration of software infrastructures to mitigate effects and impose best practices.

The third paper on "Automatic Generation of E-Government Forms from Semantic Descriptions" by Bernd Stadlhofer and Peter Salhofer, shows how an ontology for specific public services can be used as the basis for the automatic generation of web forms – in this way, ensuring that the data is automatically associated with a computable representation of its semantics, from the point of acquisition onwards. Indeed, we can be sure of the semantics of data in advance of its acquisition, and use this information in the design of subsequent processes for analysis and integration.

The fourth paper titled "Semantic Frameworks for e-Government" by Charles Crichton, Jim Davies, Jeremy Gibbons, Steve Harris and Aadya Shukla explains how a practical, semantic framework can be defined in terms of: terminology services, metadata registries, and model repositories. The first presents a collection of defined terms, structured in a way that suits many applications. The second presents a collection of 'metadata elements' - templates for recorded data, explained partly by reference to the controlled terminology, and organized in terms of equivalence, specialization and versioning. The third consists of formal models, ontologies, and metamodels describing components, processes, and information artifacts, such as database schemas, service descriptions, forms, queries, and reports. The attributes of these artifacts – fields on forms, or columns on spreadsheets – are defined, and given a clear, computable semantics by linking them to the elements of a metadata registry. Coupled with a model-driven approach to development, this allows the production of systems for electronic governance in which interoperability, and consistency with a formal specification, can be guaranteed.

7. ACKNOWLEDGMENTS

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

8. REFERENCES

[1] Ogbinaka, O.M. "A Course Text on Government for Schools, Colleges and Universities", ISBN 978-35675-0-0, 2002.

[2] Peristeras, V. and Tarabanis, K. "Advancing the Government Enterprise Architecture – GEA: The Service Execution Object Model, EGOV 2004, LNCS 3183, pp. 476-482, 2004.

[3] Traunmuller, R. and Wimmer, M.A. "E-Government at a Decisive Moment: Sketching a Roadmap to Excellence", EGOV 2003, LNCS 2739, pp. 1-14, 2003.

[4] Lenk, K., Traunmuller, R. and Wimmer, M. The Significance of Law and Knowledge for Electronic Government, Gronlund, A. (Ed.), Electronic Government: Design, Application and Management, pp. 61 – 77, 2002.

[5] Konstantina, K. Stefanos, G. and Konstantinos, M. "Towards a Privacy Audit Programmes Comparison Framework", R. Traunmuller (Ed.): EGOV 2004, LNCS 3183, 2004.

[6] Oteniya, G., Estevez, E., Zhen Zhong, D., Ojo, A. and Janowski, T. Electronic Delivery of Social Welfare Services – Development Document, e-Macao Report 4, October 2005.

[7] Costa, R. A., Leitao M. J. and Vila Verde, I. "Electronic Voting: An All-Purpose Platform", Wimmer, M.A. et. al. (Eds.): EGOV 2005, LNCS 3591, pp. 309-316, 2005.

[8] Sonntag, M. and Wimmer, M. "Legal Aspects of one-Stop Government: The Case of Applying for a Building Permission", EGOV 2003, LNCS 2739, pp. 385-392, 2003.

[9] Khadroui, A., Arni-Bloch, N., Leonard, M. and Ralyte, J. "Law-Based Ontology for E-Government Information System", MATIS, University of Geneva.

[10] Janowski, T. Ojo, A. and Estevez, E. The State of e-Government in Macao, e-Macao Report 2, April 2005.

[11] Ojo, A., Oteniya, G, Chau, K.F., Estevez, E. and Janowski, T. Electronic Delivery of Licensing Services – Development Document, e-Macao Report 5, October 2005.

[12] Oteniya, G, Estevez, E. Dai, Z.Z., Ojo, A. and Janowski, T. Electronic Delivery of Social Welfare Services – Development Document, e-Macao Report 4, October 2005.

[13] Krachina, O., Raskin, V. and Triezenberg, K. "Reconciling Privacy Policies and Regulations: Ontological Semantics Perspective", Human Interface and Mgt. of Information. Interacting in Information Environments, LNCS 4558, 2007

[14] OMG, "Unified Modeling Language 2.1.1", www.omg.org/technology/documents/modeling_spec_catalog.htm , 2007.

[15] Hoare, C.A.R. "Communicating Sequential Processes", Prentice Hall International, 1984-2005.

[16] Happel, H.J. and Seedof, S. "Applications of Ontologies in Software Engineering", 2nd Int. Workshop on Semantic Web Enabled Software Engineering (SWESE 2006).

[17] Barnett, M., Rustan, K., Leino, M. and Schulte, W. "The Spec# programming system: An overview", In CASSIS 2004, LNCS vol. 3362, Springer, 2004.

[18] Barnett, M., DeLine, R., Fähndrich, M., Rustan, K., Leino, M. and Schulte, W. "Verification of object-oriented programs with invariants", JOT 3(6), 2004.

[19] Cavarra, A., Welch, J. "Behavioral Specification from Class Models", Integrated Formal Methods, LNCS 4591, 2007

[20] Top Quadrant, "Semantic Technology", www.topquadrant.com/documents/TQ04_Semantic_Technology_Briefing.pdf, 2004