Formal Methods for Electronic Government

Jim Davies and Jeremy Gibbons

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford OX1 3QD, UK
http://www.comlab.ox.ac.uk/people/{Jim.Davies,Jeremy.Gibbons}/

Abstract. Electronic government is a challenging domain for software engineering, with complex requirements involving agility, transparency, accuracy, and accessibility. The techniques of *semantic frameworks*—metadata-based, model-driven development—may help to address these challenges. Data semantics and model transformations are prime application areas for formal methods, and so electronic government is an exciting new domain for education and training in formal methods.

1 Introduction

Increasing reliance upon electronic communication, together with the ambitions and demands of a global information society, means that electronic government is becoming the expected means of implementation for government policies, activities, and initiatives. Although considerable progress has been made, the reputation of public sector information technology remains poor. Most people can quote at least one high-profile disaster, in which a large electronic government project singularly failed to deliver.

The challenges in developing information technology for public sector applications are in principle no different from those encountered in other large, enterprise computing initiatives. They are, however, exacerbated by three main factors: the likelihood of conflict and misunderstanding between different stakeholder groups; the fact that requirements are linked to changes in policy and legislation; and the expectation that data and processes should be accessible, and also compatible with those in other initiatives.

We believe that a big step towards addressing these challenges can be made by integrating ideas from *data semantics* and *model-driven development*, an integration we call a *semantic framework*. Moreover, we claim that semantic frameworks both provide an interesting new domain for, and can derive great benefit from, work in formal methods. This paper sets out our position.

1.1 Electronic government

The term *electronic government* means more than a literal translation of existing government services and processes into electronic form: it carries expectations of transformation, often in connection with hopes for a better society. Issues such as access, transparency, change, democracy, and interaction, suggest that there may be specific domain challenges in electronic government, with significant implications for software design and development. In particular, electronic government requires a significant degree of formalisation and computerisation of semantics. The size of the community, the rate of evolution, and the importance of documentation make it essential that the semantics can be accessed, maintained, and incorporated into delivered systems without the need for extensive, error-prone manual intervention.

1.2 Challenges

The requirements of electronic government systems are more complex than those of their commercial counterparts; they are also more subject to change. Policy reforms or shifts in public opinion may require substantial changes to the design of a system, changes that may be expensive to make once implementation is under way. The government of a developed country may be able to afford such costs, but the government of a developing country cannot. In a commercial context, it is quite common to find that information system design is shaping business processes; in electronic government, this is less likely to be acceptable.

It is also more important that these requirements are correctly reflected in the behaviour of the system. In electronic government, computing systems will do more than facilitate policy—they will serve as its principal, and perhaps its only, implementation. This has significant implications not only for the criticality of development processes, but also for the design of the systems themselves.

In a commercial system, the information pertaining to an individual may define and constrain that individual as a customer; in a government system, it may define and constrain that individual as a citizen. The data may be driving the processes of government as they act upon the individual: there is a greater responsibility to maintain its correctness and availability over time. After all, most commercial organisations have competitors, and a dissatisfied customer may always change provider; that option is not nearly as straightforward when the provider is the customer's national government, with a monopoly on their relationship.

In electronic government, the stakeholders, including the end users, have a particular relationship to the processes of development and operation: this system is being procured, designed, developed, and operated on their behalf, and at their expense. We might consider there to be an implicit contract, reflected in the system requirements, similar to that which exists between government representatives and the people they represent. This means that the extent to which requirements are 'owned by the users' is far greater, and thus the system must be a better fit for the social processes that it is intended to support, than is often the case in ordinary enterprise computing. Furthermore, stakeholders may require more in the way of evidence that the system is in fact doing what is expected—the implicit contract applies in operation as well as in development.

2 Formalisation

The large-scale sharing and integration of data from dynamic, heterogeneous sources requires computable representations of the semantics of data, and it is here that a significant part of the challenge lies. Natural language or informal understanding is sufficient for such a semantics only when the concepts are straightforward, the community is small or homogeneous, and the period of time over which understanding must be maintained is short. For complex problems, heterogeneous communities, or long-running initiatives—all characteristics of electronic government—a more formal approach is required. The semantics has to be amenable to automatic processing, and this processing has to be automatically linked to the processing of the data itself. This entails the faithful representation of *data semantics* in constructing models, and the application of *model-driven development* in generating system artifacts—queries, scripts, programs, services, forms, and interfaces—from these models.

2.1 Metadata-based

Robust, trustworthy, and transparent information systems require the careful consideration and representation of the semantics of the information they record; a structured, computable representation is essential if we wish to adopt and maintain rich terminologies across multiple initiatives. Conflicts and misunderstandings about the semantics of data can be resolved, or at least identified at an earlier stage, if aspects of structure, functionality, and interpretation are conveyed through the use of models. This is standard practice in software engineering; however, the audience for the model is usually quite restricted, and thus much of the detail, or semantic metadata, may be left implicit. For electronic government, it is a requirement that models may be validated, so that public servants can be held accountable; it is therefore more important that the models are comprehensive, and that metadata is properly recorded.

2.2 Model-driven

The dynamically evolving context of policy and legislation, the greater requirements for accountability and transparency, and the sheer scale of many electronic government initiatives, all encourage the automatic generation of a system implementation from more abstract models. Information systems have always been modelled, but often only informally, using fragments of specification, written in natural language, and presented as reports, spreadsheets, and diagrams. These are partial descriptions, often containing apparent contradictions, and there is no prospect of using these to generate a system automatically. Yet these are the documents that inform decisions such as those on whether to proceed, on project scope, on supplier selection, and on contract fulfilment, and it is here that a semantic framework can start to produce real benefit. Reports and spreadsheets in which key terms are annotated with a link to agreed terminology, and data elements are annotated with a link to detailed semantics in a metadata registry, can be concise and unambiguous, while making explicit a shared understanding of exactly what is required.

In development, more formal models—typically, object models and service descriptions—can present precise descriptions of structure and functionality in which data attributes have an accessible, computable semantics, and terms have an agreed meaning. It may then be possible to determine programmatically—at the design stage, or after deployment—whether two systems are holding data that has exactly the same semantics. This is an essential prerequisite to the systems integration required for 'joined-up government', in which central and local government, different departments and agencies, work together to tackle social problems.

One way to represent the semantic information required, and to facilitate programmatic access, is to represent the various aspects of semantics using models of usage. We can identify three particularly useful kinds of model: *ontologies*, models which explain the meaning of a metadata item in terms of named relationships to other elements; *applications*, models in which the item appears in context: for example, in the context of a design document, or a form template; and *transformations*, models which explain how data collected against one set of elements can be transformed to fit another. Although only the first of these is usually seen as defining or recording meaning, the others also have semantic import: meanings are sometimes best expressed, and will evolve, through usage.

2.3 Semantic frameworks

The ideas of metadata-based and model-driven development together make what we call a *semantic framework*. A practical semantic framework can be defined in terms of constructs at three different levels: *terminology services*, *metadata registries*, and *model repositories*. The first level presents a collection of defined terms, structured in a way that suits one or more possible applications. For example, a terminology for education might include terms such as 'institution' and 'qualification', record that the terms 'university' and 'high school' denote particular kinds of institution, and record also that the terms 'master's degree' and 'international baccalaureate' are related in some way to the notion of institution.

The second level presents a collection of metadata elements, each of which describes a measurement or observation. A metadata registry for education might include elements such as institution attended, full title of degree awarded, and result obtained. Each element may be related to one or more terms in the underlying terminology, and additional semantic information is provided by informal explanations of intended purpose and an association with a domain of possible values. The registry also records relationships between elements, such as equivalence, specialisation, and versioning.

The third level presents re-usable models for the definition of information artifacts, such as database schemas, service descriptions, forms, queries, and reports. A model repository for education might include models of admissions forms, study transcripts, and spreadsheets for reporting registration and progress data to national agencies. The fields on the forms, the entries on the transcripts, and the columns on the spreadsheets may be described, and given computable semantics, by linking them to the metadata elements in a metadata registry.

In the Software Engineering group at Oxford, we have explored these ideas in the domain of clinical trial informatics. The *CancerGrid* project is a consortium involving the universities of Oxford, Cambridge, Birmingham, Belfast and London, funded by the Medical Research Council with additional support from Microsoft Research. For the last three years, the consortium has been developing a common vision for semantic frameworks and model-driven software engineering, focussed upon software support for the design and operation of cancer clinical trials. We believe that the ideas are more widely applicable than clinical trials, or even than health informatics; indeed, we believe that they are a close fit for the challenges of electronic government.

3 Education and training

Formal methods have traditionally been seen as most applicable in limited domains: typically high-integrity, safety-critical, embedded systems. Electronic government represents an exciting and important new application domain, and an opportunity to widen the impact of formal methods: the challenges of representing data semantics — precisely enough to support the automatic generation of the information systems that manipulate that data — call for the leverage that only formal methods can apply.

Electronic government exemplifies what is becoming known as the *digital economy* [2] — the transformative effects of technology upon society. Successful developments in such domains necessarily entail a multidisciplinary approach, taking into account issues of management, user engagement, ethics, security, and society, as well as the more obvious technical matters of computer science. The leaders of the digital economy must be broadly educated: it is more important that they have some appreciation of all of these issues, than that they are a specialist in one. We see the digital economy as a promising initiative for widening the scope of education and training in formal methods.

4 Acknowledgements

This position paper draws on an earlier paper [1], with contributions from Aadya Shukla and Steve Harris, and also on long and detailed discussions on the Digital Economy with Marina Jirotka, Janet Smart, and others at Oxford.

References

1. Charles Crichton, Jim Davies, Jeremy Gibbons, Steve Harris, and Aadya Shukla. Semantics frameworks for e-government. In Theresa Pardo and Tomasz Janowski, editors, *International Conference on e-Government*. ACM, December 2007.
2. EPSRC. Centres for doctoral training in the digital economy. `http://www.epsrc.ac.uk/PostgraduateTraining/NewCentres/DigitalEconomy.htm`, March 2008.