5. System and design issues (Statistics Canada)

back to case study home page

5.1 IT Architecture

Statistics Canada is moving towards a SOA. A key enabler of SOA is the Enterprise Application Integration Platform (EAIP) that allows the delivery of solutions based on meta-data driven, reusable software components and standards. Most business segments will benefit from the common core business services, standard integration platform, workflow and process orchestration enabled by the EAIP. The platform also simplifies international sharing and co-development of applications and components.

Web services currently in use and under development by EAS are associated to information objects representing *core business entities* (e.g., *que stionnaires*, *classifications*, *tax data*, *business registry*) that are classified into GSIM's *Concepts* and *Structures* groups. This fits nicely with GSBPM as well: services provide the inputs and outputs to GSBPM statistical processes. They satisfy a basic set of SOA principles, i.e., they are loosely coupled (consumer and service are insulated from each other), interoperable (consumers and services function across Java, .NET and SAS), and reusable (they are used in multiple higher-level orchestrations and compositions). Work continues to establish a complete framework, including discoverability (via a service registry and inventory) and governance.

At this point, Statistics Canada has a combination of services and silo-based/point-to-point integration that can be described as a combination of maturity levels 3 and 4 in terms of the Open Group Service Integration Maturity Model (OSIMM) maturity matrix (see Figure 1). During the transition years to a corporate-wide SOA, incremental changes are being made by applying SOA adoption and governance by segment in which cross-silo services and consumers coexist with point-to-point integration of systems and data. Early adopters of SOA services include IBSP, SSPE and SNA.

Developing Data Service Centres (DSC) is a key initiative that fits into Statistics Canada's emerging SOA. The objective of the DSC is to manage statistical information as an asset – to maximize its value by improving accessibility, utility, accuracy, security and transparency through the use of a centralized inventory of statistical data holdings, associated metadata and documentation. Key statistical files and associated standard metadata (i.e., file name, type, description, creators, owners, etc) will be registered and integrated into statistical processes via SOA. This integration will rely on a data access layer with common interfaces to access statistical files without the user needing to know their location, format and/or technology.

5.2 Metadata Management Tools

IMDB metadata discovery is performed via a Wiki-based solution and MetaWeb. Each Wiki page provides the context of the information and all available links. These pages are programmatically generated based on templates developed for the IMDB. MetaWeb is a JSP and Servlets-based application. Data are collected and populated into the IMDB via a Microsoft Excel IMDB Extraction/Loader, an Oracle PL/SQL IMDB Loader and MetaWeb.

The starting point for the Common Tools project (See Section VII - Figure 2) is the Questionnaire Development Tool (QDT) used to enter specifications for social survey data collection instruments. All question metadata is entered in the QDT, including questions and answer category text, interviewer instructions and conditions controlling flows. The Processing and Specifications Tool (PST) then loads variable metadata such as variable name, length and type. These are linked to question metadata already entered via QDT so no re-entering of question or answer category text is required. Finally, the Social Survey Processing Environment (SSPE) utilities use collection layouts or schema to generate variable metadata to be loaded to the metadata repository. Two projected tools will complete the picture: the Data Dictionary Tool (DDT), which will provide an interface to the metadata repository for updating descriptive variable metadata, and the Derived Variable Tool (DVT), which will allow entry of specifications for derived variables and will be used to produce detailed documentation for data users. Within Statistics Canada's SOA, the SSPE metadata repository will export metadata in a canonical model to IMDB via an EAIP service under development[1] .

Solutions and tools are needed to support other types of metadata, specifically in the GSIM *Structures* and *Production* groups.

---

[1] See Section IV-F for more information on SOA.

5.3 Standards and formats

The following is a list of standards and formats and where they are being used:

- BPMN – EAIP orchestrations;
- ISO/IEC 11179 Metadata registries – IMDB;
- CMR – IMDB;
- DDI 2.1 – DLI and Research Data Centres;
- DDI 3.0 – IMDB tool (automate metadata “wrap” for microdata files/PUMFs) and web services (extract metadata from IMDB for DLI and Research Data Centres). See Section IV-E for more details on this project;
- ISO/TS 17369 SDMX ML – Formatted data from dissemination;
- Neuchatel Terminology Model Part 1;
- Classification database object types V2.1 – Standards Division;
- ISO 3166-1:2006 Part 1: Country – Standards Division;
- ISO 19115 Geographic Information – Geography Division;
- ISO 15489-1 Part 1: General – Information management.

5.4 Version control and revisions

For web services that expose information assets, not only the underlying data evolve (both in content and structure) but also the services that expose it. As a result, (potentially) different versions of the same data will be published and exchanged by (potentially) different versions of the same service. No centralized versioning framework for data exists and many areas have customized versioning schemes.

For example, the IMDB allows time travel by version and effective period. A new version of a metadata item is created by copying an existing item, making necessary changes and assigning the version number to the immediate next version. Each version has an interval of validity (or effective period) associated to it. In other words, the lifespan of each version of a metadata item can be determined; conversely, the version of an item in effect at a specific point in time can also be determined.[1]

Service versions are identified using a three-digit versioning scheme: *major.minor.patch*. An increment in the *major* version requires some of its consumers to change their code. This happens because of a major change in the service contract, e.g., at least one operation has been removed or an operation signature has changed in a way not foreseen by the extension points defined in the Web Service Description Language (WSDL) file.[2] An increment in the *minor* version does not require changes on the consumer applications. These are implementation changes and/or backwards compatible changes to the interface, e.g., additions of operations or extensions to data types in the WSDL file. An increment in the *pat ch* version is only used for bug fixes.

Service versions are designed with the goal of making them as *forward* and *backward* compatible as possible. By making the interface extensible, *forward compatibility* makes room for future, uncertain functional requirements. This approach is guided by knowledge and best practices in SOA interface design, XML schema design and type theory (since forward compatibility of service interfaces is essentially a special case of subtyping). *Backward compatibility*is achieved in the usual way: by ensuring that consumer applications developed for older versions of the service can continue to work with the new version.

---

[1] Not every change to a metadata item generates a new version: versioning of different entity types (surveys, classifications, questionnaires, etc) are handled by a different set of business rules.
[2] Interface specification that describes the functionality and data types of a web service.

5.5 Outsourcing versus in-house development

External consultants were contracted for building DDI services and tools, specifically to develop in-house DDI expertise and a set of core SOA web services around the IMDB. These services expose IMDB content in a standard format compliant with the DDI XML specification to support applications that focus on different types of metadata (e.g. surveys, variables, classifications, concepts, etc.). Rather than integrating with the IMDB on a case-by-case basis (point-to-point integration), the web services enable applications to gain access to its content in a standard based format. This initial effort defined and implemented a core metadata service that delivers IMDB content encoded in DDI XML. A testing tool was also developed based on a set of common use cases (see Figure 3) to validate the effectiveness of the approach. The service is used to support the Data Liberation Initiative (DLI) and the Canadian Research Data Centre Network (CRDCN) Metadata projects comprising 25 Research Data Centres (RDCs) from universities across the country. The services were developed with a Java technology stack, including some JPA components for database access that were reused in other in-house services.

In addition, EAS developed a proof-of-concept client based on JSPs, Servlets and XSLTs to transform and render the DDI XML content returned by the data service into human-readable HTML and other proprietary formats for interoperability with internal applications (e.g., SQL Server, SAS).[1]

---

[1] See Section VII – Figure 4 for the overall architecture of the IMDB DDI services.

5.6 Sharing software components of tools

Statistics Canada's emerging SOA is providing the next generation of software components to be shared across the Agency. Services are reusable: they are designed to be combined with other services to create more complex solutions. In addition, generalized systems are being wrapped with a service interface to increase interoperability by shielding users from older technologies and multiple platforms.

One of the main challenges of this approach is that the same abstract information object (e.g., questionnaire, classification, T1 tax data) can be physically implemented by different data producers (and even by different data consumers) in different ways. This "impedance mismatch" has historically been addressed by *point-to-point data integration*, i.e., either the producer or the consumer has to conform to the other's data model. With SOA, canonical information models are created to which both producers' and consumers' models will *map* (*SOA data integration*). Canonical information models are enterprise-wide, common representations of information objects – a sort of "lingua franca" for data exchange. These models enable the organization to share and exchange enterprise information that is consistent, accurate and accessible. A mapping is a specification that describes how concepts from two different models relate to each other. At the physical level, it actually specifies how data are *tr anslated* between two models. Canonical models are not intended to *replace* the disparate set of heterogeneous physical models in use across the organization. Prescribing a single model would be impractical and counterproductive. Instead, both data consumers and producers can continue to use their own models (relational database schemas, SAS files, etc.) within their own environments and just map to the canonical only when data need to be exchanged.

Within the SOA framework, canonical models are implemented as object models that are serialized into XML Schema Definition (XSD) types. Data producer and consumer schemas are mapped to the canonical object models used by services via schema mappings – object-relational (ORMs) or object-XML (OXMs). An inventory of canonical XSD types is currently being created; it can be referenced and reused by multiple service contracts (WSDL) in the EAIP schema registry. These XSD types will be maintained by the service developers within the governance framework set up by the EAIP.

When exchanging data from a source database to a consumer application, there are a number of mappings involved along the way. First, data need to be extracted from a relational or multidimensional database into the canonical object model. This could be done automatically by object-relational mapping (ORM) tools, when the source schema is close in structure to the canonical, or it may require customized SQL/MDX extraction queries. At the other end of the process, the canonical object model is serialized into XML/JSON to be shipped to the client application via a web service interface. This mapping is done automatically by the EAIP tools. Finally, the client application needs to map the XML/JSON produced by the service into its own object model via an automatic de-serialization process. This process may include some XSLT transformation when the canonical model is very different from the consumer model and requires restructuring.[1]

(a) Example: Classification service

Classifications were one of the first core business entities to use an EAIP service. The Classification canonical model is based on GSIM and Neuchâtel. The first version contains the basic classes needed to support a classification structure, namely Scheme, Levels and Items. Each Scheme consists of one or more Levels (i.e., classes), each of which consists of one or more Items (i.e., members). This model will be extended to include Versions and Variants as necessary.

To expose IMDB data in this canonical model, the IMDB's ISO/IEC 11179 Metadata Registries entities need to be mapped to GSIM/ Neuchâtel. The IMDB data model does not have Scheme, Level and Item concepts (at least not with the usual GSIM/Neuchatel semantics), so a mechanism identifies and extracts them from the IMDB physical model via SQL mappings. At the conceptual level, this can be done by defining *containment* mappings that are expressed as subtypes between both models.[2]

There are parent-child hierarchies defined on Classification Level and Classification Item. The Level hierarchy is linear (each level has at most one child) and the Item hierarchy is a tree (each item may have zero or any number of children). Both hierarchies are related by a constraint that ensures that two items are in a parent-child relationship only if their respective levels are in a parent-child relationship as well. This constraint ensures that both hierarchies remain consistent.[3]

---

[1] Section VII - Figure 5 depicts the entire process of exchanging data from a source database to a consumer application.
[2] Section VII - Figure 6 shows the relationship between both models (the canonical entities are those starting with the word "Classification"). For the purpose of defining a mapping, Classification Schemes and Levels can be viewed as a subtype of Enumerated Value Domain (EVD), whereas Classification Items are a subtype of Permissible Value (PV). Classification Schemes are a special type of EVD with no PV (i.e., Item) directly associated to them – Items are only associated to Levels. All Items associated to a given Level have different Values.

[3] Section VII - Figures 7 and 8 show the actual physical mapping for Classification Scheme, Item and Level. The mapping is defined by UML notes (the boxes with the bended corners). The syntax of the mapping is straightforward: the "<<" symbol indicates an assignment from the attribute on the right to the attribute on the left. In addition, there are constraints on code sets from the IMDB content model.

5.7 Additional materials

----[1] Daniel W. Gillman, 1999: Corporate Metadata Repository (CMR) Model; U.S Bureau of Labor Statistics.