

# 5. System and design issues (Australian Bureau of Statistics)

## 5.1 IT Architecture

ABS Enterprise Architecture harnesses The Open Group Architecture Framework (TOGAF) which recognises domains of business, data, applications and technology architecture. In describing "IT Architecture" below, reference is primarily made to applications and technology architecture. Connections with data architecture are also explored.

Unless otherwise noted, descriptions in this section refer back to the main metadata systems as described in Section 4.1.

The newer metadata facilities are based on a Service Oriented Architecture. The older facilities tend to have monolithic coupling of the repository, the business logic and business rules (which are built into the application rather than embedded in services) and the User Interface.

Nevertheless, selected information about the collections defined in CMS is "projected" from CMS into an Oracle database. While only a small subset of the total information held in CMS, this comprises all of the core "structural" registration details about collections, cycles and profiles. Basic (read only) "collection metadata services" based on this content on Oracle are then provided for statistical processing applications to access.

A similar approach applies in the case of classifications except a much greater percentage of the total information held in regard to classifications is both "structural" and available on Oracle.

Apart from CMS and ClAMS (which include some descriptive content held only in IBM's Lotus Notes product) the other metadata holdings are all based in Oracle. There is extensive use of Oracle Stored Procedures for reusable services/functions and some use of true web services.

In summary, more recently developed facilities based on recent architectural standards within the ABS, tend to consist of

- a store (typically Oracle based)
- that is wrapped with low level Create, Read, Update, Delete (CRUD) services
  - "D" often actually refers to "Deprecate" (eg marking metadata as no longer the version recommended for use) rather than physically deleting metadata.
- that, in turn, are used as building blocks for higher level "business services" related to the store
  - these business services which consistently enforce business logic/rules - including resolving on an authenticated roles basis who is permitted to do what in terms of CRUD operations on specific elements of content within that store of metadata
- there is typically also a generic GUI associated with the store, for general browsing, management and administration purposes
  - typically, however, most business applications (eg for statistical processing and dissemination) simply access and apply the business services in the manner they require to interest with the metadata content rather than making use of the generic GUI
  - external applications are not able to use SQL or other means to interact with the metadata content store except via the CRUD layer

While SOA offers a lot of opportunities and potential, it also comes with a lot of new complexities compared with earlier approaches. It requires new understandings and a new mindset from those developers who are being asked to take up, and interact with, the available services as well as requiring the same from the business analysts and programmers within the team responsible for providing the metadata repositories and services. It can make the overall environment more complicated in some ways (eg services are calling services that call services etc and then somewhere at a low level a service is updated and everything needs to be configured appropriately to allow proper testing of that change). Implementing SOA in environments that include a lot of "legacy" processing systems that are not enabled for the new architectural directions is particularly challenging

During 2008 it became clearer that a significant aspect of the work on establishing an updated and coherent metadata framework for the ABS amounts to defining Enterprise Information Architecture (EIA) in the context of a statistical organisation. Without a clear and coherent EIA, there is a risk each service, or each bundle of services, is delivered with its own explicit or implicit information model. The ABS could have gone from having a dozen or so environments with subtle and not so subtle differences in their underpinning information concepts and structures to having an array of services based on a plethora of different, and unreconciled, information models. On the positive side, SOA can help make EIA practical and consistent. Rather than having the same objects and relationships specified in the EIA implemented, and extended, differently across a number of different environments, a single consistent but flexible bundle of services could be used within each environment. SOA and EIA are complementary rather than alternative directions.

The IMT strategy addresses the requirement for SOA and EIA to work together. It enables common information constructs, defined according to schemas aligned with relevant standards such as SDMX and DDI, to be used consistently via service layers. These service layers enforce core business rules. They also mean application developers can work with information objects at a business level without needing to understand, and code based on, the full details of the SDMX and DDI information models. The integration with Statistical Workflow Management is also an important element of the "to be" IT Architecture.

## 5.2 Metadata Management Tools

Statistical processing applications interact with metadata via services where possible. As described in [BHM](#), however, many ABS processing applications and third party vendor products are not yet amenable to this approach. Where this approach is used currently it most often involves the application "reading" relevant content from the metadata repository rather than writing back new or updated records.

The [IMT](#) strategy seeks to fully, and consistently, realise this approach. Some existing key applications (and repositories) may need to be "wrapped" so they can interact with the MRR on a CRUDS basis. ("S" refers to harnessing the MRR Search capabilities to support discovery, selection of relevant content to Read etc.). Other legacy applications may need to be decommissioned, through delivery of services and interfaces that take their place, and content from a number of legacy repositories will need to be migrated to the (logically) centralised repositories associated with the MRR.

In the meantime, as described in the introduction to 2.2, there are cases where metadata from the Corporate Metadata Repository needs to be restructured and/or repackaged relatively manually to make it suitable for use in particular processing systems.

## 5.3 Standards and formats

Standards and formats currently in use for major metadata repositories are described in Section 4.1.

Under [IMT](#), the primary standards are SDMX and DDI, interoperating with other "purpose specific" standards such as

- ISO 11179 for concepts.
- ISO 19115 (and related standards such as ISO 19139) for geospatial metadata, together with relevant OGC (Open Geospatial Consortium) standards for geospatial data and registries.
- Dublin Core and related standards for discovery metadata.
- BPMN for process modelling
- BPEL for process execution

Regardless of which standard's information model is being harnessed, content for interchange (eg to be read by applications) is typically represented in XML. In order to reduce the need to exchange large XML structures, where only a small proportion of the total information may be needed for a particular application, the XML used to describe an object can refer to sub components and related objects "by reference" rather than including all this information "in line". The calling application can then resolve the specific references (if any) which are relevant to its particular needs – once again typically resulting in smaller packages of XML than would be the case if a comprehensive set of information related to the component was included "in line".

While XML is used for interchange, current repositories tend to store content using RDBMS (relational database) technology. XML stores and graph databases are technologies being considered for future to augment RDBMS approaches.

Expression in RDF format (which builds on simple XML representation) is seen as an important additional capability in future. This is seen as one advantage of harnessing standards – in many cases the community for a standard has already developed a recommended expression in RDF.

## 5.4 Version control and revisions

The approach to versioning has been a major point of debate within the ABS previously. As the systems have grown up at different times, their approach to version control tends to differ.

In general, where there was not seen to be a compelling case for supporting formal versioning past developments tended to avoid that "complexity". Collections, for example, are not currently versioned. Many aspects of change over time for a collection, however, can be handled through descriptions of the "cycle" or the "profile" rather than edits to the main collection document itself.

Under [IMT](#), however, versioning is seen as a prerequisite for active use and reuse of metadata. The structural definition of a metadata object at the time it was referenced must remain accessible even if a new version of that object is defined subsequently. This is consistent with the approach taken in standards such as SDMX and DDI. Both of these standards have a concept of objects being able to be in "draft" mode in which case they should not be referenced for production purposes. The standards do not require versioning of drafts but it is likely that the MRR will support versioning of drafts.

Past debates over when a change is so fundamental that it should result in definition of a new object, rather than a new version of an existing object, remain to be addressed in the [IMT](#) context.

Past debates about changes that are so "trivial" (eg fixing a spelling mistake) that they shouldn't result in version change also remain to be finalised in the [IMT](#) context.

An example of problems from lack of appropriate support for versioning in current infrastructure is classification system. It could benefit, for example, from the [Neuchatel](#) approach to modelling classifications, versions and variants as well as the [IMT](#) approach to not overwriting previous content.

Within the current system each registered object is essentially an independent entity (ie a "new classification"). It is possible to designate one classification as being "based on" another but this can mean many different things

- The new classification is a new version of the earlier classification and is in some sense expected to supersede it (although possibly not

- immediately).
- The new classification is a "variant" of the earlier classification defined for a specific purpose. The earlier classification may "live on" indefinitely for the original purpose.
- Classifications are being "grouped" into a "family" without necessarily being formal variants or versions of each other.

Where revisions are to be made (or new versions created) as much impact analysis as possible is undertaken. This includes, for example, understanding what other metadata objects and processes refer to the object that is about to be revised (or versioned) and whether the revision will have any inappropriate impact (whether the new version should be referenced instead). The lack of fully "joined up" registries (including knowing exactly what metadata is referred to in each processing system) makes impact assessments difficult and only partially reliable in some cases.

The MRR and Statistical Workflow Management working together in future should greatly assist in this regard. While existing metadata objects and business processes will be able to continue referencing the present version of an object that is proposed to be updated/versioned, understanding these existing uses and the requirements associated with them

- may assist in designing the new version of the metadata object to best address "whole of business" needs
- will allow the full set of users of the existing version to consider whether they should now use the new version or continue using the present version

The preceding example illustrates the flow on impacts that versioning can have within a complex and actively used metadata registration system. If the existing metadata objects that refer to the object that just got "versioned" now need to refer to the newer version of that object, all those existing metadata objects themselves now need to get "versioned" (because they're pointing to a different version of the first object). All the objects that refer to the objects that referred to the original object now need to get assessed and potentially versioned themselves, and so on with a ripple effect potentially sweeping across the whole registry originating from just one object being versioned. (While standards such as DDI-L support the option of "late binding", they recommend against it for many purposes. Under "late binding" a reference to another object is always deemed to refer to the most recent version of that object – rather than, eg, to the specific version of the object that was current at the time the reference to it was made. "Late binding" reduces precision and leaves open the possibility that the object referred to will subsequently "evolve" in ways that contradict the initial basis for referring to it.)

The IMT approach supports user decision points (which may be manual or automated) in regard to the "ripple effect" of versioning. It also provides the greatest systematic support for managing initial and "consequential" versioning processes.

## 5.5 Outsourcing versus in-house development

While external expert consultants were engaged from time to time, the existing metadata systems described in Section 4.1 were all designed and developed "in-house". Open source and other starting points for the Data Element Registry were seriously considered.

ABS (and the Australian Government) ICT Policy and Strategy is placing a greater emphasis on COTS (Commercial Off The Shelf) & GOTS (Government Off The Shelf) based. "Bespoke" software developments (whether through in house development or commissioning of external developers) to deliver all, or part, of a solution is seen as a last resort if other options are demonstrated not to be viable.

From an ABS perspective, however, it remains typically the case that in house staff

- assemble the functional and non requirements that need to be met by a new solution
- identify solution options and assess them against these requirements
- manage the integration and commissioning of the new solution within the existing IT and business environment

Ensuring solutions are consistent with Enterprise Architecture, including Service Oriented Architecture and support for relevant open standards, promotes effective integration (with minimum need to re-engineer other systems), reduces risks of "vendor lock" and facilitates end of life decommissioning (and possible replacement).

The approach to IMT aligns with these ICT strategies and policies. This includes

- the emphasis on sharing solutions (eg across NSIs and other agencies)
- the preference, where no existing solution is suitable, for collaborative development
  - this includes harnessing, where applicable, existing common frameworks (such as those developed within the European Statistical System) and utilities as "building blocks" for the solution
  - development of new shared solutions is seen as preferable to undertaking purely local "bespoke" developments

While not all developments related to IMT will necessarily deliver, or harness, open source components, open source is recognised as one important paradigm for sharing solutions and sustaining their evolution over time.

In addition to seeking to collaborate with other agencies, ABS is drawing on input from expert consultants to assist developers understand and apply information standards such as SDMX and DDI and to assist in designing key infrastructure such as the MRR.

Development of REEM (Remote Execution Environment for Microdata) is an example of the ABS working with a vendor that shares our interest in harnessing standards such as SDMX and DDI-L. Elements of the REEM solution include

- the ABS developing services to extract metadata from existing systems to populate a DDI description of unit record data
- the vendor "wrapping" existing COTS components to be able to work with DDI and SDMX, and
- components developed specifically by the ABS for purposes such as user registration

ABS implementation, as ABS.Stat, of the OECD.Stat platform is an example of harnessing an existing standards aligned shared solution and entering into a collaborative partnership (with OECD, IMF, Statistics New Zealand and Istat) to maintain and evolve that solution in future.

## 5.6 Sharing software components of tools

At present, many systems (as described in Section 4.1) used by the ABS are built in a "monolithic" fashion (combining the repository, the business logic and the user interface) and are highly customised for the ABS environment (eg they rely on both IBM Lotus Notes and Oracle databases which are configured in a particular way). CMS, ClAMS and the Dataset Registry are all in this category. While there is no in principle objection to sharing these components with other agencies, doing so in practice would be very complex both for the ABS and for the other agency. In any case, as these facilities were developed more than a decade ago and predate relevant application architecture and metadata standards, it is not anticipated any other agency would be interested in making use of these facilities in their current form.

Newer facilities such as the Data Element Registry (DER) and Questionnaire Development Tool (QDT) are architected in a manner that would make it easier to share them. Both of these facilities are designed so that a user interface interacts with the Oracle database via a "Business Services Layer" (BSL). In addition to full sharing, partial sharing could be supported (eg the ABS providing the repository and BSL, with the other agency choosing to develop its own user interface.)

Sharing could be envisaged in at least two forms. One would be the ABS packaging either the full facility or some layers from the facility in a form which allowed another agency to establish a "stand alone" instance. A second form would be extending the BSL (and probably repositioning the repository) so that authorised and authenticated interactions from outside the ABS became possible in regard to the current instance of the facility. One or more external agencies might then act as registration authorities in their own right. This could have many benefits in terms of sharing, and shared development of, metadata content but would be likely to require more thought in terms of ongoing governance and support arrangements.

A third possibility, which physically "cloned" the repository (ie the first option) but supported a unified logical perspective across the original repository and the clone(s) (ie elements of the second option) would also require significant additional work.

While these facilities are deliberately more compartmentalised and self contained in design, they were not developed from the ground up with the intent of sharing beyond the ABS. Some generalisation of ABS specific aspects (eg linkages of both the DER and QDT to collection information from the CMS) would still be required.

The software the ABS has available should be able to be made available to other statistical agencies free of charge in its current form. If the ABS needed to modify the software and/or provide consultancy support in order for that software to be made operational outside the ABS then that work may need to be cost recovered. Alternatively, and preferably, it may be possible to agree a collaborative arrangement such that the existing facility is extended and generalised in a manner that benefits both the ABS and the other agency.

The ABS seeks to avoid becoming a "software house". Any sharing arrangements would be in the context of either one off provision or, preferably, some form of partnership. A relationship along the lines of the ABS acting as a provider to one or more "customers" does not fit with current ABS aspirations and directions.

A number of other ABS applications (eg ABS Autocoder and REEM) are also listed in the Sharing Advisory Board's [inventory of software available for sharing](#).

Short of sharing software itself, the ABS is very happy to exchange details of data models, application architectures, user experiences etc with other statistical agencies.

New developments such as the MRR are being designed to be more readily sharable, in whole or part.

While the ABS has relatively few components currently that other agencies may be interested in sharing, the ABS is placing a very high priority on establishing collaborative partnerships with other agencies to develop new components, or to extend existing modern standards aligned components that already exist outside the ABS.