

Distr.
GENERAL

CES/SEM.47/3
21 January 2002

ENGLISH
ENGLISH and FRENCH only

**STATISTICAL COMMISSION and
ECONOMIC COMMISSION FOR EUROPE**

**COMMISSION OF THE
EUROPEAN COMMUNITIES**

CONFERENCE OF EUROPEAN STATISTICIANS

EUROSTAT

**Joint UNECE/Eurostat Seminar on Integrated Statistical
Information Systems and Related Matters (ISIS 2002)**
(17-19 April 2002, Geneva, Switzerland)

Topic I: Application of web technology to integrate statistics

USE OF A WEB PORTAL TO ACCESS COMMON REFERENCE DATABASES

Invited paper

Submitted by Statistics Canada¹

Abstract: This paper will present an architectural framework for integrating reference databases under a common web portal. The architecture focuses on *coherence* rather than function. It assumes that basic functions of existing components need to be preserved while new processes and services must be provided across the enterprise. The proposed framework deals with removing definitional inconsistencies and providing a seamless service interface to several groups of users. The architecture presented provides an approach to managing the key reference sources within a statistical organisation.

I. INTRODUCTION

1. In developing this architecture, some choices have been made from a technology perspective; they reflect a widespread trend within the industry and demand from users to employ a web-based approach. The assumption is that most services will be delivered to the desktop using an Enterprise Intranet, and accessed with a common browser. The framework also uses a commercial technology known as *Enterprise Application Integration (EAI)*, which is described later in the paper.

2. The examples used in this paper to illustrate the architecture and its concepts are mostly drawn from the administrative domain, rather than from statistical processing. These were used for reasons of simplicity and common understanding. The application of the architecture to statistical databases is discussed in the last section.

¹ Prepared by Mel J. Turner (Mel.Turner@statcan.ca).

3. It is important to point out that this paper represents an architectural approach that may be followed in the future, rather than a description of a system that has actually been implemented. Like many organisations, Statistics Canada is undergoing a transition to web-based technologies yet must preserve its investments in existing applications and data. We are also confronted with the availability of many new commercial products that provide pieces of the solution, and the emergence of technology “standards” at an unprecedented rate. It is to this rapidly changing context that the architecture seeks bring some coherence.

A. What is a Web Portal?

4. For the purposes of this paper, a web portal is a web site that acts as a gateway to a set of related information and services. It is designed to serve a defined community of users. The *gateway* metaphor implies a measure of security, or access control, meaning that users would have to log-in when they arrive and their identity would influence the nature and availability of services provided.

5. We use the term “web” in the context of the technologies generally associated with the Internet or World-Wide Web. However, this paper deals exclusively with internal services that might be implemented on an Enterprise Intranet, rather than provided for external users.

B. Users and Services

6. Web technologies are becoming the preferred solution to deliver services to communities of users that are widely dispersed in the organisation. The overriding advantage for users is the availability of the services from any desktop using a common browser. For the system implementers, this means a much simpler way to distribute software capability when compared to the previous generation of client-server systems.

7. To develop the architecture in this paper, certain basic assumptions have been made regarding the users and the range of services that are provided through the web portal. These are as follows:

- ?? Users may be anywhere in the organisation, provided they have access to an intranet-connected workstation.
- ?? Users are named and have valid computer user IDs. They are required to log-on to the web portal before being given access to the services. Session “cookies” are used within the web browser to control access and are set to expire after 10 minutes of inactivity; this requires a user to re-establish the session if they leave the workstation.
- ?? Users have different *roles* such as: employee; manager; administrator or developer. Their current role is used to customise the web session, exposing only relevant services. Users each have a default role that is remembered from session to session and have to explicitly change roles if necessary.

C. Architecture Objectives

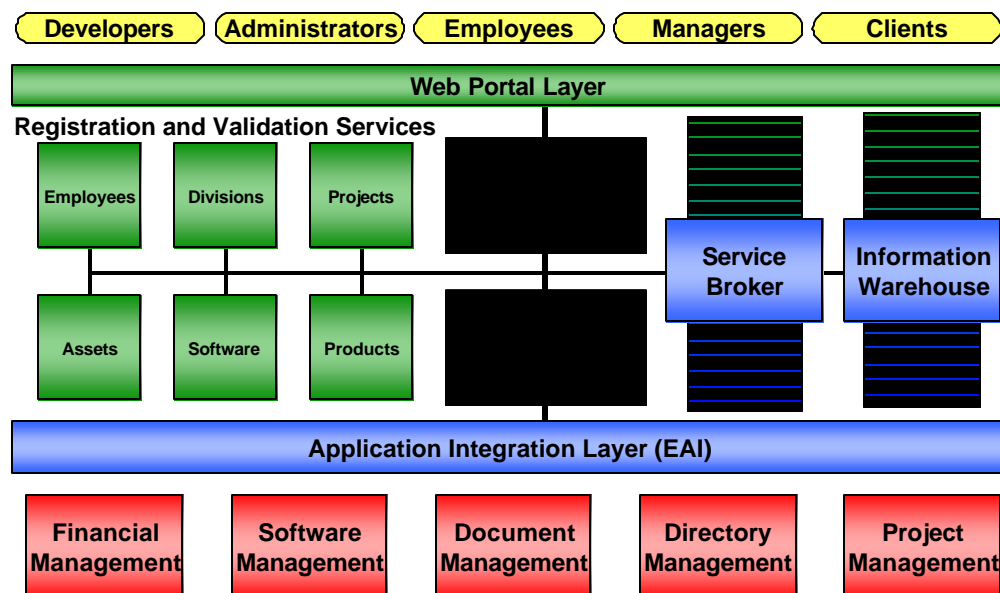
8. The architecture explained in this paper is a high-level *reference architecture*. Its objective is to provide a conceptual foundation for design, rather than specific recommendations for products or implementation methods. The main issue addressed by the architecture is the integration of existing applications, many of which use different technologies and incorporate data that is inconsistent, while adopting a common service interface.

9. From a user perspective, it is important to remove the inconsistencies and provide a unified view of the information and services available. Where possible, the integration should enable users to work more effectively by automating tasks, eliminating duplicate data entry and smoothing the workflow between applications.

10. From a development perspective, the objective is to explore the opportunities represented by new commercial technologies, particularly the class of products referred to *enterprise application integration (EAI)* and the middleware infrastructures on which they depend. The architecture also touches on *data warehousing* issues because, in many ways, the integration of multiple sources of data for reporting purposes shares the same problems of harmonising concepts and information.

11. It will come as no surprise that the key to integrating application services and information is meta-data. Within a web portal, meta-data of many different kinds is integral to the operation and functioning of the system itself, providing technical specifications as well as information directly to users. The paper explores the new demands placed on meta-data by this architecture.

II. AN ARCHITECTURAL FRAMEWORK



A. Architecture Overview

12. The framework shown above illustrates the principal components of the architecture for integrating the typical administrative services found in any enterprise. At the bottom of the diagram are blocks representing existing applications and databases, often managed independently by different divisions within the organisation, that actually provide the services and manage the administrative information.

13. The framework deals with the use of *meta-data* and *registration* processes that ensure a single integrated view is provided to users, even where there are inconsistencies in the underlying databases. The combination of registers and other meta-data provide the conceptual integration or harmonisation of data held separately by the existing application components.

14. The main purpose of the architecture is to provide services to the groups of users accessing the web portal. This is achieved by the *Service Broker* component that has access to a special kind of meta-data shown as *Business Rules* in the diagram.

15. The final integration component in the architecture is the *Information Warehouse*. This is a replacement for the myriad of reporting modules that characterise the existing systems. These must be

replaced because they expose to the user the incompatibilities and inconsistencies of the underlying systems. A warehouse is a specially designed database of microdata drawn on a regular basis from the existing, separate data stores. However the data it contains must be fully rationalised with the user-visible meta-data and the contents of the registers. In other words, the consistency often has to be engineered by reformatting and mapping the underlying data into a common form.

B. Identity Management

16. The first issue in providing a unified view to users is equivalent to a data modelling activity in application design: what are the primary entities participating in the services to be provided? In the administrative example followed by this paper, these are employees, projects, services, divisions and other assets of the corporation that must be referred to by name (or other identity) when users request information or services. These identities become the *nouns* in any sentence describing a service instance, such as: “Assign *employee-name* to this *project*”.

17. The portal architecture has a database of *identity registers*² that is enterprise-wide in scope and not part of any of the existing applications that are being integrated. The purpose of these registers is to verify the legitimacy of any identifier used in user dialogues, and consequently in any transaction that is issued to an application. In addition the registers contain the fully dependent attributes of these identifiers that are required for meaningful user communication. For example, an *employee identifier* is usually a meaningless unique number that is replaced in a user dialogue by their name.

18. The portal will become the steward of these identities at the enterprise level, even though one or more of the applications will probably duplicate this function. The portal also provides *registration services* that update the registers and initiate transactions to keep the legacy databases synchronised.

19. Identity management provides an example of an important architectural principle regarding the underlying legacy databases:

*The portal may have direct **read** access to the databases of legacy systems but does not update them directly. Instead, it may issue asynchronous messages or transactions to the legacy applications for later action. The only databases directly updated are those owned by the portal, consisting of registers, meta-data and business rules.*

20. There may be rare exceptions to this read-only principle but, as a general rule, it is required to ensure the portal is scalable to many users and not invasive to the legacy applications. This also implies that some services provided through the portal are time-dependent because the actions are deferred until the legacy system is scheduled.

21. The portal registers also play a role in *privilege management*, determining the services that users may access and the information content they may see. We recommend consideration of the Registry Information Model proposed by Gallagher and Carnahan³ that provides a comprehensive registry design that includes the tracking of roles and privileges and accommodates the registration processes envisaged by this architecture.

2 Identity is the basis of structural integration (whether assembling individual legal entities and operating entities into a business enterprise, or assembling individual metadata concepts into a classification system).

3 A General Purpose Registry/Repository Information Model; Len Gallagher, Lisa Carnahan; October, 2000; National Institute of Standards and Technology.

C. Metadata Management

22. The integrated meta-database shown in the diagram is another kind of portal register. Like the others, it contributes primarily to validate the vocabulary of the user dialogues when using the portal. In this case the focus is on data elements, concepts and classifications, together with their representations within the portal and in participating application systems. The meta-database is also used to describe and control several aspects of the portal itself, including the entry forms, labels, selection lists and descriptive help information needed for the user interface.

23. When the metadatabase can be updated dynamically by a user registered as a *developer*, the portal can be used directly to define new business services that can subsequently be made available to other users.

24. In object oriented terms, the meta-database defines the *types* of objects and their associated characteristics while the identity registers define and track the *instances* of these objects.

D. Service Brokering

25. The combination of identity registers and meta-data owned by the portal is sufficient to validate a wide range of transactions that are destined for the back-end legacy systems. Many of the conditions that would cause a transaction to be rejected by the back-end applications can be pre-checked and corrected by the user. In addition, meta-data that describes transaction formats can be used to properly format or transform data so it is suitable for each application.

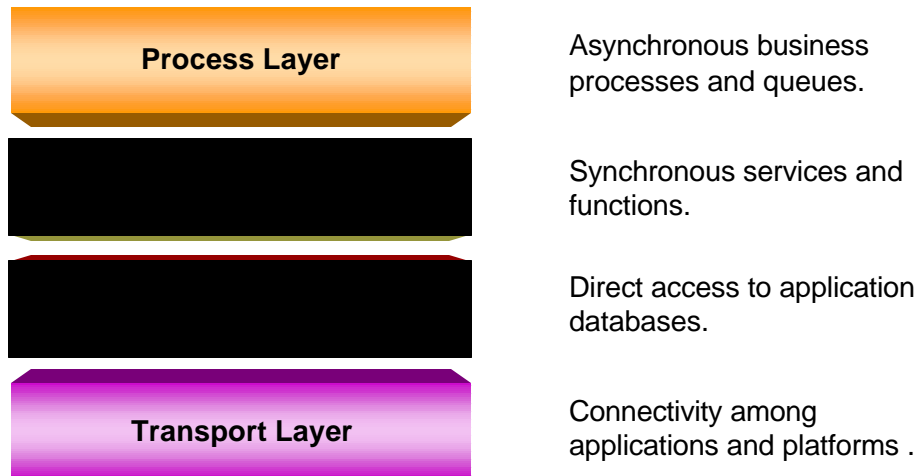
26. The portal may also contain a repository of *business rules* as a special kind of business data. These rules are used to control sequencing within the portal and how the services provided by back-end applications are provided to users. More detailed examples of brokering are provided below under *Application Integration*.

E. Warehouse Management

27. The final service class that may be provided by the portal is support for accessing and reporting the data maintained by the operational data stores within the back-end applications. In most implementations, the data warehouse is independent of the legacy systems but is built from periodic extractions from the operational data stores that have been harmonised in terms of data formats and concepts. The warehouse must be entirely compatible with the meta-data held within the portal so that the users receive a coherent view. In simple queries, this transformation could be performed dynamically but this is not usually feasible for more sophisticated summarisation and reporting.

III. APPLICATION INTEGRATION

28. Given a common approach to *identity management* and a consistent repository of meta-data, we can turn our attention to the applications within the organisation that will be integrated and exposed to the users of the web portal. These are often legacy applications, built using a variety of technologies and on multiple platforms. In general, they were not designed for web interaction and are likely not consistent with each other regarding their definitions of business entities and processes.



Application Integration Layers

29. Integration can occur at several levels among applications and may require significant effort to engineer. The model below identifies four logical levels, each of which may require different technical solutions:

- Transport Layer:** This is a foundation layer that serves to harmonise the communications protocols and provide connectivity among applications and platforms. As well as basic protocols, such as TCP/IP, this layer also includes synchronous message brokers (RPC, COM, CORBA) and asynchronous message queue infrastructure (MQSeries, mail messaging).
- Data Layer:** In some cases it may be appropriate to interact directly with the application databases without involving the existing applications themselves. Because the application's business rules are bypassed, it is often necessary to restrict this kind of access to "read-only" so that the database integrity is preserved. This layer uses standard protocols such as SQL and common interfaces such as ODBC.
- Function Layer:** Ideally, applications expose "callable" interfaces that provide complete business functions to the web portal. By "complete" we mean the interaction is meaningful in business terms (e.g. Add a customer) and that all business rules and constraints that preserve the integrity of the application are applied. These functions are usually synchronous; meaning the processes that invoke them can determine their success or failure.
- Process Layer:** This layer deals with the sequencing of functions and services within the architecture. It is hierarchically an abstraction layer⁴ above the applications being integrated and is the principal topic of this paper. It is served by *Enterprise Application Integration (EAI)* technology that manages workflows and messaging across applications as well as serving the user interactions provided by the web portal.

30. Given this basic model, a relatively simple scenario of integrating applications within the web portal architecture is described below. The example deals with the administrative functions associated with a new

⁴ Traditional development practices distinguish build-time and run-time phases. Newer practices distinguish build-time (component building), deploy-time (dynamic assembly) and run-time phases.

employee joining the organisation, and providing that employee with the services required to function in their job. This involves interactions with a number of applications located in different operational areas, and these are not currently integrated.

A. Requirement

31. The business requirement is to provide new employees with a single point of contact when they join the organisation, and to be able to complete the usual administrative processes using Intranet services.

32. The separate applications involved are as follows:

- ?? Building security: In order to obtain a photo ID for subsequent access to the building the individual's security credentials are verified and an identity card is issued. This process is common to all people needing regular building access, including employees, contract personnel and building maintenance workers.
- ?? Computer access: Employees and contract workers are provided access to the computing infrastructure by issuing a user ID and registering this ID with the e-mail system, perhaps an Internet account and other privileges.
- ?? Employee registration: For employees only, the administrative systems handling pay and benefits must be notified.
- ?? Services registration: The employee may be optionally registered for a variety of services such as Library privileges, Training, etc. Preferably, these secondary registrations can be accomplished directly by the employee, once the above credentials have been established.

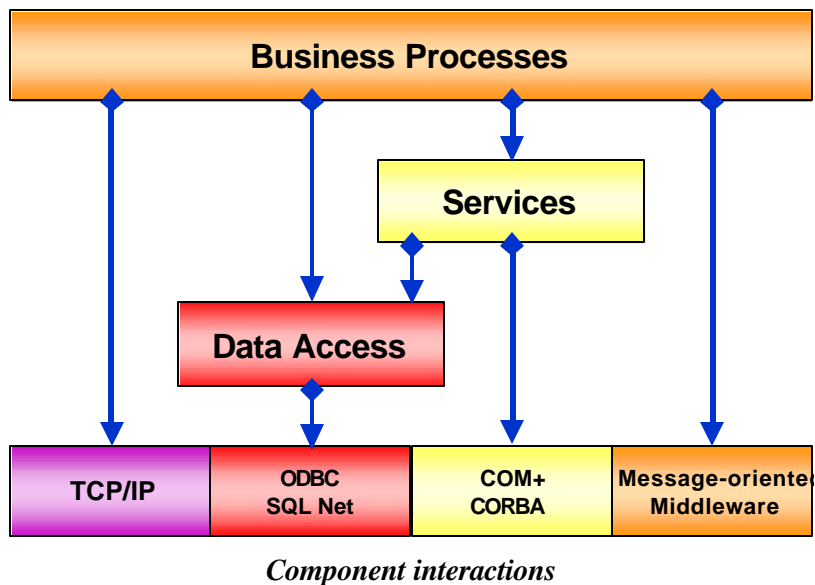
33. In this example, the *building security* system is outside the web portal architecture except that its database is exposed for read-only access via the enterprise infrastructure. It will be involved as an example of *data integration* using an ODBC interface. Only basic personal details are captured when issuing a building pass such as contact information that has been manually verified against a driving license or passport. For privacy reasons, no linkage with these other identifiers is retained but a unique ID number is issued and encoded on the photo identity card.

34. An administrator, who has access to the web portal, can carry out the process for registering the employee. Usually the business process will be designed so that the employee themselves can be responsible for entering and updating most of the administrative information required. However, until the employee has been granted a computer user ID, they are unable to access these services.

35. The administrator makes use of the portal registration service to begin the process. The employee must show their building pass (to verify their identity) and ID number is entered or captured by a card reader. This number is verified as legitimate against the building security database and initiates registration. The issuance of a computer user ID is an example of a function that can be directly invoked by the web portal in real time using a COM connection to an operating system function. That is, it is an example of *functional integration*. The operating system includes all the logic to issue the ID, record an initial password, and update the underlying directories within the infrastructure. The portal layer retains the ID in its registration database if it is successfully issued.

36. The portal registration service also has to notify the administrative pay and benefits system of the new employee but this system is less tightly coupled than the computer registration process. In this case, an on-screen form is used to capture the necessary information (pre-filled if the information is already available to the portal) and a transaction is constructed to send to pay and benefits. This transaction is queued by the

EAI product and will be delivered reliably when the pay and benefits system is ready to receive it. In reality, the pay and benefits system may be a legacy batch application that is invoked periodically to process any transactions that have accumulated. This asynchronous delivery mechanism is an example of *process integration* that is accomplished using message-oriented middleware solutions such as MQSeries.



37. The diagram shows a summary view of the different interactions among components. This shows a number of sample technologies within the transport layer to communicate among components, all of which may be referred to as *middleware*. These various technologies may be incorporated within a single *EAI* product that could also include mechanisms to perform data transformations and business rules.

B. Data Transformation

38. The integration of applications that were developed at different times for various purposes inevitably involves a careful review of the data concepts and implementation formats. The *EAI* product should provide extensive capabilities to transform and reformat data as it is passed between different applications. It is for this reason that the Web Portal architecture includes the *Integrated Meta-Database* as a key component. The *IMDB* is the repository of data transformation rules that will translate data, moving to and from existing applications, to harmonised concepts and formats.

C. Business Rules

39. Managing business processes goes far beyond transforming data and passing it to the appropriate application for processing. Even within the very simple example above, identities must be verified and transactions sequenced and distributed to other components. This sequencing may depend upon the outcome of prior functions, or on specific values entered during a user interaction.

40. More generally, the complete *EAI* solution may include automatic responses to events. For example, applications designed to process batches of transactions may be invoked at specific times of the day, or when a volume of transactions has accumulated. This provides the loosest kind of coupling between components but avoids the processing bottlenecks that can slow online response times.

41. This event-driven type of integration operates outside the web portal architecture and is referred to as *workflow management*. The *EAI* solutions that operate in this domain use a "*publish and subscribe*" model

to trigger asynchronous processes or to route messages between processes. When the messages are complete documents or business forms, routed to human interactors with the system, the electronic mail infrastructure may be the supporting technology.

42. The web portal architecture includes a repository for business rules that operate at a more granular level, within the *process* layer. That is, they specify the sequence of decisions and actions that occur within a business process.

D. XML

43. Although application integration can be accomplished using a variety of components and messaging technologies, this requires significant expertise and results in systems that are complex and difficult to maintain. There is a strong trend in EAI solutions to reduce this complexity by specification-driven approaches and the use of graphical interfaces to design and construct solutions.

44. In describing the architecture, we have identified several different kinds of specifications. In an actual implementation these would be supplemented by definitions of user dialogues, screen layouts, report layouts, message structures and transaction formats, to name a few. To simplify this proliferation of meta-data and business rule specification, the *eXtensible Markup Language (XML)* is emerging as a common syntax in which these definitions are expressed.

45. In addition to expressing the specifications in XML, this same syntax can be used for the messages that pass between components and for the data content. That is, XML is becoming the *lingua franca* for all layers within the integration model. This has the advantages of universality and vendor independence as well as being easily communicated among platforms and across the Internet.

46. However, using a standard syntax is only one dimension of standardisation. The use of XML has given rise to the emergence of several sub-languages that deal with different aspects of handling documents, their transformation, their routing and their definition. There are also broader efforts to introduce standard definitions of the documents or messages themselves, so that inter-organisational (B2B) integration can converge on common business transactions.

47. A language relevant to the task of application integration is *XML Stylesheet Language Transformation (XSLT)*, a standard written by the Worldwide Web Consortium (W3C), which encompasses both business rules and data transformation. XSLT is a language designed to transform one XML document to another and is able to change both structure and content.

IV. APPLYING THE ARCHITECTURE TO STATISTICAL SYSTEMS

48. The examples given above are administrative in nature, and might occur in the operational requirements of any organisation. Within a statistical organisation, this architecture may also be applied when integrating the statistical infrastructure composed of registers, survey processing and respondent interactions. In many ways, a statistical infrastructure is more demanding than a typical administrative system and this section highlights some of the additional requirements that could be envisaged.

49. As an example, consider the presence of common portal for business and individual respondents who need to register themselves for electronic data reporting on one or more surveys being conducted by the statistical office. The portal does not provide any statistical processing functions *per se* but can be designed to manage the common functions of *contact management*, *response management* and *data marshalling* to back-end processing systems.

50. In this example, the portal would manage the following registers:

- *Contact Register.* This register identifies individuals who represent the respondent business or household, and who are authorised to transact data associated with a particular survey instance. In general, there may be more than one contact for a survey response, particularly for business surveys, and a given contact may be a respondent for more than one survey. The data retained within the portal for each contact is not confidential (in a statistical sense) and relates only to authentication and identification. For example, it might include *name, e-mail address* and *telephone number*.
- *Survey Register.* Each instance of a survey (cycle) that can accept electronic transactions is registered here. The register would also identify the XML definitions that are valid for this survey. That is, it identifies the electronic form representing the questionnaire and its associated meta-data. It would also identify the back-end application responsible for processing responses and other public information that might be useful for respondent contacts (standard information documents about the survey, data usage statements, privacy policy, etc.)
- *Response Register.* This register tracks response instances, maintaining a log of the interactions of a specific contact for a specific survey instance. This information is retained for survey control purposes, for example to enforce business rules concerning multiple responses or to identify a telephone follow-up action.

51. This represents a bare-bones survey portal that can manage the registration of *contacts* by a survey manager (or automated system); allow contacts to identify themselves when responding electronically to a survey instance; and then manage the sessions for that contact during the response cycle. In a simple e-mail survey, the contact register would be used when sending the electronic questionnaire to respondents (using secure e-mail if the questionnaire contained any respondent-specific data). The questionnaire might contain, for example, the *business number* as well as the *survey instance id*. These identities allow the contact to register and submit a response through the portal. The response package could be an encrypted electronic form passed directly to a back-end processing system.

52. In this example, the portal is not aware of the survey content and deals only with survey control information. The survey interaction itself is handled by an off-the-shelf e-form mechanism, secured by encryption technology. If the portal architecture is used to manage the content of response data, the following considerations are relevant.

A. Harmonisation⁵ of Concepts

53. Providing a common portal for multiple surveys (to a business, for example) will place far greater emphasis than before on the harmonisation of the data involved. As noted before, tools within the portal can provide basic restructuring and reformatting of transactions to make them suitable for interfacing to existing processing systems, but differing concepts would present a confusing interface for the respondent.

⁵ Harmonization is the human intellectual process that produces the rules for coherent assembly. These assembly rules, once defined, may be applied in constructing business enterprises out of previously identified component legal entities and operating entities. Harmonization may also be applied to the more abstract problem of designing/assembling classification systems using component classes.

54. The emphasis in harmonising content should occur first for the identification and classification information that is typically held in statistical registers such as the *business register*, *address register* and *tax-data register*. These registers contain administrative data that has been originally provided by the client (often through multiple channels or sources) or assembled and imputed by the statistical office.

55. In many countries, it is becoming a requirement under privacy legislation to expose (in an appropriately secure manner) administrative records to their owners so that they may correct any data held in error. Although statistical data is often exempted from such legislation, this exemption may not cover data collected through administrative sources.

56. Standardisation of data concepts and classifications that occur within administrative records is outside the scope of this paper but it is worth noting that XML is increasingly being used as syntax for describing public interfaces for data. For example, XBRL (*eXtensible Business Reporting Language*) is a standard under development for defining the chart of accounts and structure for company expense statements and balance sheets. If these standards attain widespread agreement and use it may become feasible to harmonise concepts, not just within a statistical office, but across multiple departments of government.

B. Synchronisation

57. The integration of statistical applications, and particularly those involving administrative registers, must resolve the conflicts caused by bringing together information from different time periods. As far as the web portal architecture is concerned, it should adopt a policy of *current time*. That is, it should expose only the most recent (or current) administrative data to its external audiences and be careful to properly label the *reference period* if the data is not current or applies to a specific period of history.

58. In general, the rationalisation of statistical data across time periods involves complex processes of calendarisation and seasonal adjustment. While such methods can improve the quality of estimates and imputation in the statistical data, the resulting values should not normally be exposed to the respondent.

59. This implies a clear distinction between the register information stored within the web portal and the usual reference databases (also often called registers) within statistical offices. Reference databases such as the *business register* hold an historical record of company structure that has varied over time. Similarly, geographic reference databases hold time-varying definitions of boundaries and administrative areas. These kinds of databases embed complex concepts that are inherently difficult to integrate and synchronise and therefore cannot be exposed directly to respondents. Rather, the web portal registers that are used to communicate with respondents should confine themselves to information that the respondent can change dynamically as events unfold in the real world.

C. Metainformation and Metadata

60. Statistical metadata also varies over time (for example, classifications that categorise industries, commodities or occupations), requiring complex re-coding or concordance exercises to rationalise data. These complexities are not appropriate for handling within a real-time web portal that should present a coherent view to a respondent.

61. This implies that the meta-data held within the portal architecture is confined to current definitions used within user dialogues, rather than the full inventory of metadata required within the statistical office.

V. CONCLUSION

62. The architecture for a web portal, as described in this paper, can be broadly applied as an interface to existing applications provided the following design principles are adopted:

- *User centricity:* The portal must appear as a single, coherent system to its users. This requires significant effort to define common data concepts and to understand the transformation of these concepts when interfacing to existing applications.
- *Simplicity:* The business logic embedded in the portal should be confined to user interaction and control or sequencing functions. Complex business rules or processes in the underlying applications should not be replicated or exposed in the portal.
- *Currency:* A web portal is a real-time interface for its community of users. Therefore it should present only current information and straightforward concepts of time.
- *Utility:* A web portal interface should present a clear set of services to its users. The kind of portal described here allows the user to be task-oriented. Services such as *registration, form-filling, submission* and *verification* can be generalised to a wide range of applications and contribute to the coherence of the user experience.

63. We have also shown that emerging standards, such as XML, and off-the-shelf technologies play a key role in developing systems of this type.

64. It is hoped that this architecture can provide a roadmap for Statistics Canada to experiment with Web technology while preserving its investments in existing applications.