Reference Architectures to Measure Data Completeness Across Integrated Databases

Nurul A. Emran[1], Suzanne Embury[2], Paolo Missier[3], and Norashikin Ahmad [1]

[1]Centre of Advanced Computing Technology (C-ACT), University Teknikal Malaysia Melaka, [2]The University of Manchester, [3]The University of Newcastle.
nurulakmar@utem.edu.my,embury@cs.man.ac.uk,paolo.missier@ncl.ac.uk, norashikin@utem.edu.my

Abstract. Completeness is an important aspect of data quality and to determine data acceptability one needs to measure the completeness of the data set of concerned. One type of data completeness measure is population-based completeness (PBC). Nevertheless, the notion of PBC will be of little use until we can determine the efforts required (in terms of architectural design) to implement PBC. In this paper, we present the types of PBC system reference architecture and motivate the selection of each.

Keywords: reference architecture, population-based completeness (PBC), data completeness, completeness measurement

1 Introduction

In measuring population-based completeness (PBC) (as proposed in [1]), one essential requirement is to identify and prepare reference population before completeness of data sets of concerned can be measured. Following the definition of PBC, a *simple ratio method* is applied to measure PBC in form of:

$$Completeness(D, RP) = \frac{|(D \cap RP)|}{|RP|} \in [0, 1], \quad (1)$$

where D is the data set under measure, and RP is the reference population.

The situation where there exists a single source that contains a good approximation of the true reference population is limited however (an exception being the Genbank database[1] that contains the genes with good evidence of their existence). Therefore, good approximate populations should be established by integrating individuals from a range of sources. To describe approximate populations established from integrated sources, we adopt the term *universe of discourse (UoD)* or in short *universe*[2]. Conceptually, a universe consists of a collection of approximate populations within an application domain used for PBC

[1] http://www.ncbi.nlm.nih.gov/genbank/

[2] The term *universe* was introduced by Augustus De Morgan in 1846 in formal logics to represent the collection of objects under discussion of a specific discourse [2].

measurements, that is built by integrating individuals from several incomplete sources for the populations.

We use the term *contributing sources* (CSs) for sources that contribute to the reference populations in the universe. The CSs could be in multiple forms, such as databases (private and public) e.g., observation databases from gene regulatory networks and pathways [3] or published literature. As it is crucial to understand (and to manage) the relationship between the CSs and the reference populations for successful integration, we propose a structure called a *population map*. Conceptually, a population map consists of a mapping between a reference population and its CSs. If a reference population is stored as a table, and the CSs are databases, we say that a population map is a mapping between a reference population table and queries over tables on CSs. Note that, as the schema of the universe may not be the same as the schema of the CSs, the designers of the population maps must consider the differences.

The notion of PBC and the elements essential to measure PBC will be of little use if they cannot be implemented in practice, in a usable system at 'acceptable cost'. The question of how acceptable the cost is, however, is subjective to the person(s) implementing the PBC system, as it cannot be assessed in absolute terms. Thus, before we can elucidate the costs involved (in terms of software and hardware components), we need to determine the form of PBC system architectures to implement.

The rest of this paper is organised as follows. Section 2 presents the conceptual elements PBC system components; Section 3 covers the forms of PBC reference architecture and Section 4 presents the implementation implications of the reference architectures. Finally, Section 5 concludes the paper.

2 A PBC System

In elucidating the efforts that are involved in developing a PBC system, we need to understand the inputs and processes that are required within a PBC system. The elements of PBC (data set under measure, reference population, universe, CSs) are the inputs of a PBC system, in which the interactions among them are made through some processes that are grouped into three main activities that any PBC implementation must support. These activities are *Setup*, *Measurement* and *Maintenance*. We created a use case diagram to elicit the use cases (that represent the main activities) of a PBC system and the possible interactions among the 'actors'. To elicit the possible interactions among the main activities, we created a main activity diagram showing the plausible flows through them. As the focus of this paper is on Setup, we will present an activity diagram created with the inputs and processes that are specific to it, where the problems and the analysis of their effect on PBC accuracy will be presented in the section to come.

We modelled Setup, Measurement and Maintenance as the main activities that must be supported by a PBC system. These activities are shown as use cases in a use case diagram as shown in Fig. 1. As a result of modelling the

main activities of PBC, we identified the 'actors' of the system. These actors are the stakeholders of the system whose interest will enable PBC system to be implemented. The actors and their interactions with the use cases are:

- Data Analyst: the end user of the system who initiates PBC measurement in order to support some data processing tasks that he or she is engaged in. For example, data analysts provide the data set to be measured and the specification of the reference population to use for measurement (e.g., the name of the reference population). This actor might also provide some inputs for the system's maintenance (such as feedback about the performance of the measurement facilities).
- Domain Expert: a person who has knowledge of the populations and the contributing sources of the populations in the application domain, who contributes to Setup and Maintenance.
- System Investor: the person who owns the system and also decides on matters relating to financial cost during Setup and Maintenance.
- PBC Measurement Designer: the person who designs the PBC measures that will be supported by the system and the technical artefacts (e.g., schema) needed to implement them. The PBC Measurement Designer engages in the Setup and Maintenance of the PBC system.
- System Administrator: the person responsible for the maintenance aspect.
- Contributing Source: the source that contributes population individuals in the universe, that interacts during Setup and Maintenance. In some cases, the contributing source might be accessed during Measurement.

We created an activity diagram (Fig. 2) in order to elicit the possible interactions among the main activities, showing the plausible flows through them. Note that [OK] and [Not OK] notations are used to indicate successful and unsuccessful termination of the activities. Among these activities, Setup is the necessary starting point that leads to Measurement and Maintenence. However, if Setup is unsuccessful, due to for example the schema of one or more CSs being inaccessible or the identified CSs no longer existing, we resume Setup until we can deal with the factors that caused the previous attempts to be unsuccessful.

We allow Measurement and Maintenance to occur concurrently as long as the Maintenance does not involve the data sets required by the Measurement. But in the case where the data sets required by the Measurement need to be maintained, the Maintenance needs to wait until the Measurement which is under way is completed. Maintenance must be resumed if it is interrupted for a reason, such as lack of storage space to add new populations.

We have seen so far the actors of the PBC system and the main activities that the actors undertake. We have also described the interactions among the main activities. However, the questions that arise are: 'What is the link among the main activities, the actors and the elements of PBC?', and 'What are the forms of architecture of a PBC system that can be used?' To answer these questions, in the next section, we will present several forms of PBC reference architecture, for which the links among the main activities, the actors and the elements of PBC will be described.

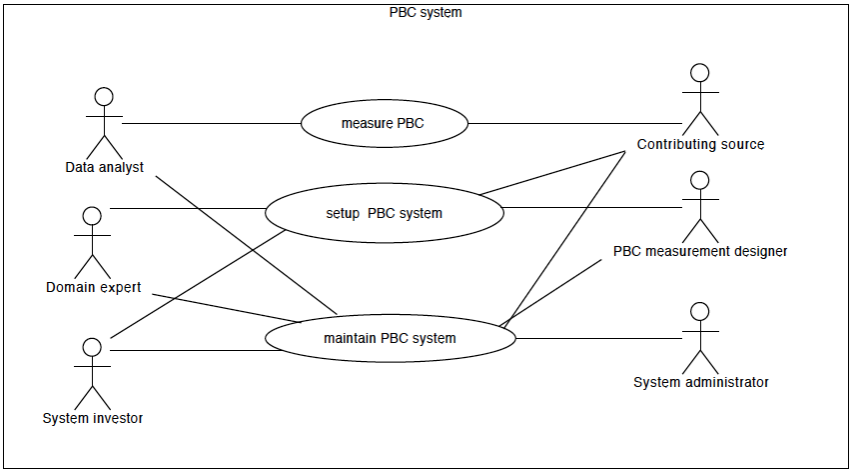

Fig. 1. Main Use Cases of a PBC System

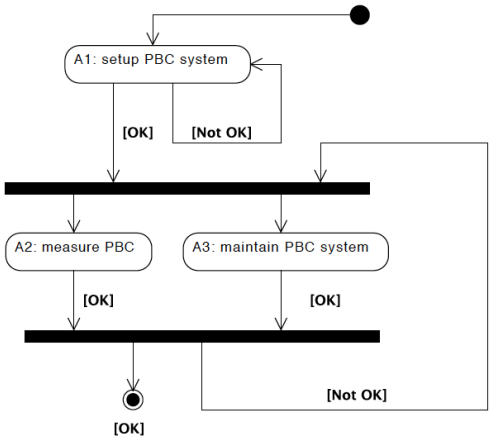

Fig. 2. Main Activities of a PBC system

3 A PBC Reference Architecture

Basically, all PBC system architectures consist of:

- Software components:
  - PBC measurement interface: a software interface for data analysts to issue PBC measurement requests (i.e. by specifying the data set to measure and the specification of the reference population to use) and to receive PBC measurement answers.
  - Measurement processor: a software program that calculates PBC measurement answers based on the PBC measurement requests by instantiating the PBC measurement formula.
  - Wrapper: software that interfaces the universe and the CSs, which is used to transform the queries against CSs that are understandable and executable by the CSs that they wrap.
- Source components:
  - Universe: a database that is either 'virtual' or 'materialised'.
  - Contributing source: source of reference populations in the universe that is in certain forms such as databases (e.g., relational and XML), published literature, or observation data (e.g., data from gene regulatory networks and pathways [TANPI09])).
  - Population map: a structure that stores mappings between populations and their CSs.
  - PBC components configuration: information about the CSs (e.g., CSs name and URI) and the universe (e.g., the reference populations it contains).

The form of the universe selected will determine the type of PCB reference architecture used and the additional software component(s) needed for the architecture.

3.1 The Virtual Universe

One type of PBC reference architecture, as shown in Fig. 3, implements the universe as a virtual universe. With the virtual universe, it means that the individuals of the reference populations are extracted from the CSs on demand. In addition to the basic components, for this form of architecture, we need a processing layer called a mediator [4, 5] that performs data integration tasks on demand. The mediator receives queries for the reference population from the PBC measurement processor (data flow a3), which instantiates the queries based on PBC measurement requests (data flow a1). The mediator executes the queries based on the mappings specified in the population maps and based on CSs information in the PBC components configuration (data flow a5) to extract population individuals from each CS through the wrappers. The queries to extract the populations are shown as data flow a6. These queries are in the form of the schema of the universe (e.g., relational query). Each wrapper will

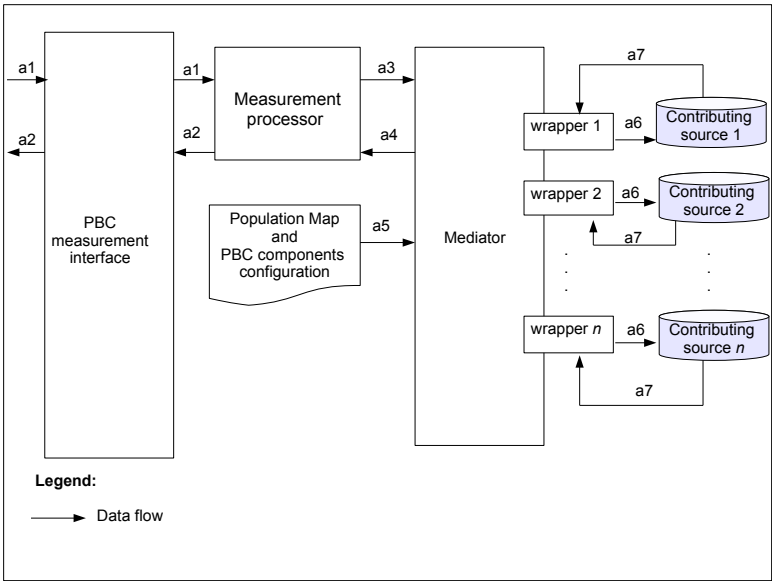

Fig. 3. A PBC Reference Architecture Using a Virtual Universe

transform the queries into the queries that are understandable by the local CSs (e.g., XML query), which relies on the type of the CSs (e.g., relational or XML databases). The wrappers also transform the query results (population individuals) into the schema that is similar to the universe's schema (called structural transformation). We may rely on the mediator to perform semantic (and format) transformation on the query results to conform to the universe schema. Population individuals from the CSs (data flow a7) are therefore integrated and transformed by the mediator, before they can be passed to the PBC measurement processor for computation (data flow a4). The results of the measurement are sent to the person requesting the measurement (data flow a2).

3.2 The Fully Materialised Universe

Another type of PBC reference architecture that we propose (based on the classic datawarehouse design [6]), as shown in Fig. 4, implements the universe as a fully materialised database. Instead of querying reference populations on an on-demand basis, the individuals of the reference populations that are extracted from the CSs (data flow a7) are stored within the universe. A similar architecture is adopted by a yeast datawarehouse for datamining purposes [7]. For this type of architecture, we need a software that could support up-front extraction of population individuals from multiple CSs, prior to the actual PBC measurements. The extraction process relies on the mappings between the populations and their CSs (retrieved from the population maps), and the information about PBC components configuration (data flow a5). The extracted data sets must

also conform to the schema of the universe, before they can be loaded into the universe. Therefore, in addition to the basic components, we propose to include an ETL[3] pipeline within the architecture that will support the tasks just mentioned. The wrappers are plugged into the ETL pipeline and the functionality of the wrappers in this architecture is the same as the wrappers in virtual universe reference architectures. The Transform component is responsible for performing semantic (and formatting) transformation and for integrating the population individuals, which is the same functionality that is performed by the mediator. The Load component loads the integrated reference populations into the universe, through the universe management interface. In this architecture, we also propose to include a software interface, called the universe management interface so that system administrators can interact with the ETL pipeline e.g., in receiving the transformed population individual data sets (data flow a8). Information about modifications on the CSs (data flow a9) is needed in order to perform updates on the universe.

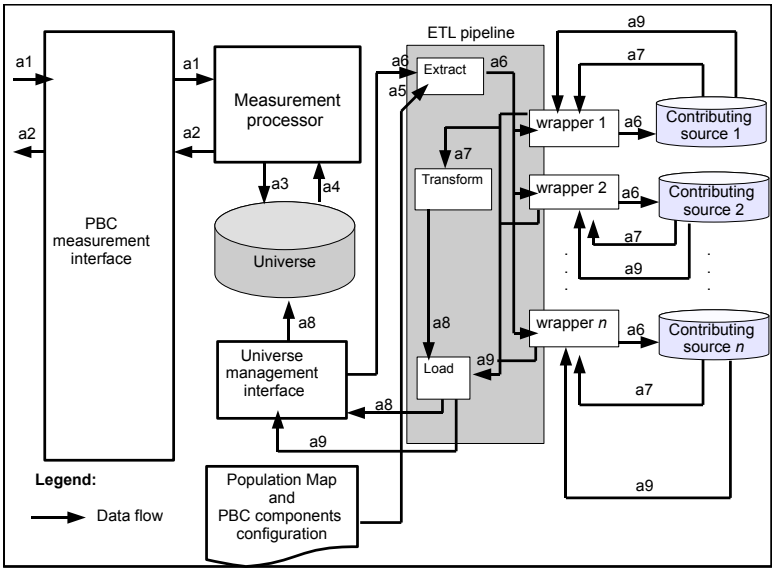

Fig. 4. A PBC Reference Architecture Using a Fully Materialised Universe

3.3 The Partially Materialised Universe

A reference architecture that implements a partially materialised universe, known as a hybrid approach [9, 10], as shown in Fig. 5, is another type of architecture

[3] Extract, transform and load. A software tool that is usually used in data warehousing and decision support system [8].

that is plausible for a PBC system. In this architecture, only some reference populations are stored in the universe while the others are kept in their own CSs and queried on demand through a mediator. A PBC reference architecture with a partially materialised universe must support queries for reference populations (data flow a3) against both the universe and the CSs. We propose to include a software component in this architecture, called the 'broker', that will decide whether queries for the reference population should be sent to the universe or to the mediator (or both). The broker must know which reference populations are materialised and which must be fetched on demand. The measurement processor will interact with the broker in issuing the queries instantiated based on the PBC measurement requests. Data flows involving the virtual universe (through the mediator) are illustrated in blue for a clearer distinction between the functionalities of fully materialised and virtual universe architecture within this hybrid architecture.

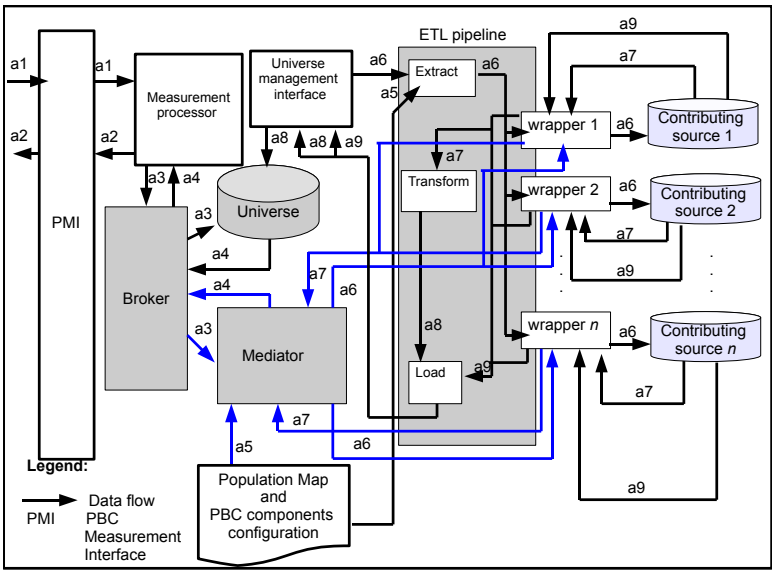

Fig. 5. A PBC Reference Architecture Using a Partially Materialised Universe

4 Implementation Implications

PBC measurement providers might choose one particular form of PBC reference architecture to implement PBC for a number of reasons.

The architecture with the virtual universe might be chosen because the reference populations are queried on-demand; therefore actual load of data is unnecessary except during the measurement. As a consequence, small data storage

allocation is needed to store the system's components configuration information (such as for the CSs configuration) and population maps. However, the virtual universe is unlikely to be very efficient if PBC measurement needs to be performed frequently, and the number of CSs that must be queried to perform the measurement is large. This is because there is a risk of communication failures, especially if the CSs are prone to experiencing technical problems that cause restricted access. In addition, PBC measurement is likely to be slow, as all the complicated tasks of querying the population individuals from multiple CSs, transforming the query results to conform with the universe schema, and combining the query results must take place before the measurement processor can initiate the PBC computation.

PBC reference architecture with a fully materialised universe serves as a better alternative if quick PBC measurement is needed. Even though an up-front effort to construct and populate the universe is needed, it could be worthwhile if a large number of data analysts are accessing the reference populations, each wishing to make use of a slightly different combination of the data set under measure and the reference population. However, we need to deal with data freshness, an issue that is common for any approach that materialises copies of data sets from other data sources. Therefore, it is necessary to put effort into detecting or monitoring updates to the contributing data sources, to ensure the materialised data sets are up to date [11].

With additional knowledge regarding which reference populations are frequently accessed (or are not frequently accessed), one might consider PBC reference architecture that implements a partially materialised universe. In terms of the requirements, a partially materialised universe needs support for the functionality required by a fully materialised universe and a virtual universe.

Regardless of the reasons for choosing a particular type of PBC reference architecture, the components in all architectures have their own functionality to support PBC measurement providers to answer PBC measurement requests. The data flows in the architectures are present as the result of the interactions among the components of the system (software and source) that each serve their own functionality (e.g., measurement processor to accept PBC request, instantiate the query and pass it to the universe/mediator). In comparison, with the fully materialised universe, a PBC reference architecture consists of all data flows that are present in the architecture that implements the virtual universe (data flow a1 to a6) with two extra additional data flows (data flow a7 and a8); all data flows that are present in the partially materialised universe architecture can be found in the fully materialised universe architecture. Using the data flows to indicate the results of the functionalities performed by the components, we say that the differences (in terms of functionality) among the three forms of architecture are small. As the functionalities required by the architecture with a fully materialised universe cover the functionalities of the other two forms of architecture, in the rest of this thesis, we will focus on a PBC system with a fully materialised universe.

5 Conclusion

In this paper, we proposed three types of PBC reference architecture, each with extra components that must be added along with the basic components of PBC. The choice of the type of PBC reference architecture to use will guide PBC measurement providers to determine the software (and hardware) components that are needed and therefore to make necessary estimates on the financial costs incurred. However, the question that remains unanswered is regarding the efforts needed to support PBC measurement providers in providing accurate PBC measurement answers. In our future work, implementation of these framework will be done to validate the hypothesized implementation implications.

Acknowledgments. The authors would like to thank the financial assistance provided by the Universiti Teknikal Malaysia, Melaka (UTeM) and the Ministry of Higher Education, Malaysia during the course of this research.

References

1. Emran, N., Embury, S., Missier, P., Muda, A.: Measuring data completeness for microbial genomics database. In: Proceedings of ACIIDS 2013. Lecture Notes in Computer Science, Springer (2013)
2. Maddux, R.: The origin of relation algebras in the development and axiomatization of the calculus of relations. Studia Logica **50** (1991) 421–455
3. Tiffin, N., Andrade-Navarro, M.A., Perez-Iratxeta, C.: Linking genes to diseases: it's all in the data. Genome Medicine **1** (2009) 1–7
4. Lenzerini, M.: Data integration: a theoretical perspective. In: Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of database systems (PODS), ACM (2002) 233–246
5. Wiederhold, G.: Mediators in the architecture of future information systems. Computer **25** (1992) 38–49
6. Inmon, W. In Elliot, R., ed.: Building the Datawarehouse. Volume 3rd edition. Wiley Computer Publishing (2002) 1–427
7. Balakrishnan, R., Park, J., Karra, K., Hitz, B., Binkley, G., Hong, E., Sullivan, J., Micklem, G., Cherry, J.: YeastMinean integrated data warehouse for Saccharomyces cerevisiae data as a multipurpose tool-kit. Database: The Journal of Biological Database and Curation **2012** (2012) 1–8
8. Chaudhuri, S., Dayal, U., Ganti, V.: Database technology for decision support systems. Computer **34** (2001) 48 –55
9. Hull, R., Zhou, G.: A framework for supporting data integration using the materialized and virtual approaches. SIGMOD Records **25** (1996) 481–492
10. Hull, R.: Managing semantic heterogeneity in databases: a theoretical prospective. In: Proceedings of the Sixteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), ACM (1997) 51–61
11. Widom, J.: Research problems in data warehousing. In: Proceedings of the Fourth International Conference on Information and Knowledge Management, ACM (1995) 25–30