



# A Logical Pattern for Integrating Business Intelligence into Information Systems Design

Stephen Opoku-Anokye, Yinshan Tang

## ► To cite this version:

Stephen Opoku-Anokye, Yinshan Tang. A Logical Pattern for Integrating Business Intelligence into Information Systems Design. 15th International Conference on Informatics and Semiotics in Organisations (ICISO), May 2014, Shanghai, China. pp.358-367, 10.1007/978-3-642-55355-4\_37 . hal-01350943

**HAL Id: hal-01350943**

**<https://inria.hal.science/hal-01350943>**

Submitted on 2 Aug 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

# A Logical Pattern for Integrating Business Intelligence into Information Systems Design

Stephen Opoku-Anokye and Yinshan Tang

Informatics Research Centre, Business Informatics, Systems and Accounting, Henley Business School, University of Reading, Whiteknights, Reading, RG6 6UD, UK  
{s.opoku-anokye@pgr.reading.ac.uk, y.tang@henley.ac.uk}

**Abstract.** In this paper, we review, literature concerning the relationship between the requirements for information systems design and business intelligence (BI). This is to provide a basis for discussion on the need for the integration of BI into information systems design. Literature on the current design patterns for information systems and BI were reviewed to identify design trends that are contributing to the use of BI that are based on only the data extracted from source information systems. We observe three main layers of logical design pattern for computer-supported information systems, namely, information presentation layer, domain model layer and information source layer. We classified according to the design purpose of each layer, which are, purposeful use of information presentation (Pragmatics design layer), meaning and understanding of information within particular business domains (Semantics design layer) and storage of signs for information captured and encoded into data (Syntactic design layer). We propose LOPIBIS as a logical pattern for integrating BI into information systems design, which is to support the use of a single version of business rules (SVOBR) to capture, process and recall same sets of data in source information systems. Subsequently enabling to the use of BI that are based on not just the data extracted from source information systems, but data with the contexts of business actions.

**Keywords:** Information Systems Design, Business Intelligence, Logical Design Pattern

## 1 Introduction

Traditionally, the design efforts for a computer-supported information system (CSIS) focus on how to provide support for a more efficient and productive business operations [1] or appropriate and effective decision making [2]. The issues considered during the initial design of CSIS are largely the collection, processing, display and storage of information for purposeful use to support business actions or decision making. Information system design involves the gathering and analysis of business requirements to produce specifications for the information system. These specifications are used to produce abstract models for implementation to support information collection, processing, display and storage. Given the focus on either efficient and effective busi-

ness actions or appropriate and effective business decision making; the initial specifications are, largely, for the design and implementation of CSIS that aims to provide support for business actions or decision making cycles. These include the need for the design of CSIS to process, store and/or retrieve transaction information, analytical information, collaborative information, etc. Business Intelligence (BI) is usually given almost negligible consideration during the design of CSIS to support business operations, collaboration or decision making. Due consideration is given, when end-users of CSIS begin to make demands for capabilities “to explore, make sense of, and gain actionable insight into rapidly changing business ecosystems.” [3] Usually, by the time system designers begin to consider BI, a number of disparate and varied CSIS would have been in use for some time and are generating varieties of data at high volumes and velocity.

BI solutions are designed and implemented in organisations to provide support for the making of effective and appropriate business decisions or the taking of efficient and effective business actions or all. Its aim is to enable people within and across organisations to make sense of data that are being captured by the various information systems that support the process of business decision making and the taking of business actions. The objectives of implementing a BI solution in an organisation include the provision of support for “complex decision making and problem solving” [4]. This objective is achieved by enabling end-users of BI to answer fundamental business questions that the operations and decision-making cycle in every organisation depends on [5]. The primary aim of designing a BI solution can therefore be said to provide support for (1) effective and appropriate business decision making; (2) improving the efficiency with which decisions are made [6]; and (3) improving efficiency and effectiveness of business actions. In current industry practice, BI is either perceived to be a concept that is commonly used to describe “technologies, applications, and processes for gathering, storing, accessing, and analysing data.” [7] or “architecture and collective use of integrated operational and decision-support applications and data-bases” [8]. BI design, therefore, involves the extraction of data from operations, collaborations or decision support systems. The extracted data is cleansed and then transformed into information and subsequently into knowledge. This information is then presented to BI end-users as reports, dashboards, scorecards, query results or analytical models. Often, the extracted-cleansed-transformed data are first loaded into a database or databases that are dedicated for BI use only. Examples of such databases include data-marts, operational data stores, analytical data stores, some data warehouse implementations, data cubes, etc.

Transformation of data as part of the BI process is achieved with the use of business rules to extract, cleanse, consolidate, aggregate, transform and make sense of data that were captured and stored within disparate CSIS. These business rules are often different from the business rules that exist in respective CSIS where the data are processed, captured and stored. The use of one set of business rules within an information system to capture and store information, and a different set of business rules to retrieve the same set of information through BI; thus presents a real risk, where, the data retrieved via BI may not have the same context as the data captured during the processing of such information. This creates gaps between the semantics of data cap-

tured and stored and information presented through BI. The existence of semantic gaps between any two or more information sources exacerbate the “levels of deceit in data and information manipulation [9]” that may go on within and across organizations. We argue these semantics gaps could be bridged when BI is integrated into information systems designs. We define two main criteria for such integration as follows:

1. The design of CSIS does not to consider just the display, processing with business rules and storage of data, but also the interaction of business rules that generated the data.
2. CSIS are designed to recall facts with context based on data with consideration to business rules interactions.

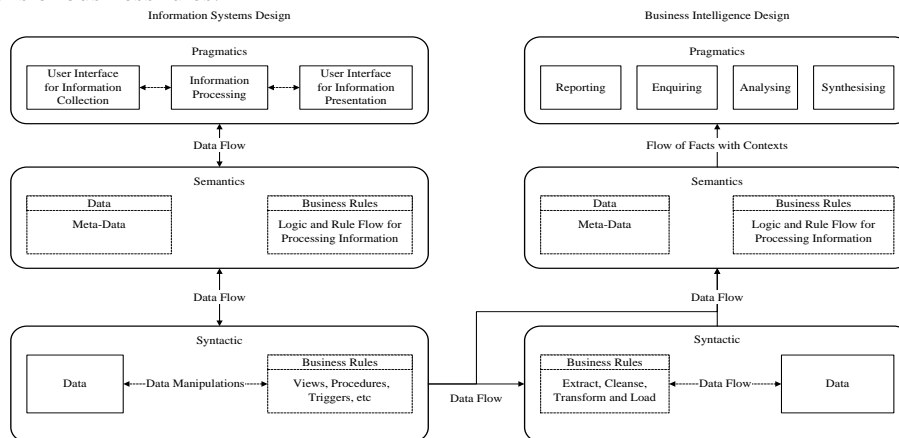
In this paper, we propose a logical pattern for the integration of BI into the design of information systems.

## **2 Design Pattern for Information Systems and Business Intelligence**

The design of CSIS includes considerations for architecture and collective use of integrated operational systems, collaborative systems, decision-support systems, databases and business rules storage systems. The physical distribution of such architectures and integrated systems are different from the understanding of their logical designs and distributions [8]. Thus the physical distribution of any information system could be different from the understanding of its logical designs and distributions. A typical internal model for a CSIS follows n-tier client-server pattern with a clear division between data and application tiers. The pre-defined business rules that are used to control data flow and processing logic usually form part of the application tier or may be separated into several layers. Sometimes, the components that make up these two tiers could be separated into several layers. [10] describe three principal layers of logical CSIS design as presentation, domain and data source. We classify each of these three logical design layers according to the design purpose of each layer and map them to the three levels of abstractions for sign-systems as: (1) a model for the purposeful use of information that are presented (i.e. pragmatic model), (2) a model for the meaning and understanding of information within particular business domains (i.e. semantics model) and (3) a model for capturing, processing, forwarding and storage of signals (i.e. syntactic model). Fig. 1 is a diagram that describes our understanding of the current logical design pattern for CSIS and BI.

Current implementations of CSIS involve the design to encode facts with the contexts of business actions into data for processing and storage. Multiple version of business rules (MVOBR) are used to capture, forward, process, encode, store, recall and decode the same sets of data. Business data are often stored in a database or databases without the interactions of business rules that were used to process and encode the information into data. The use of these data without the associated interactions of

business rule that created the data, excludes critical details about the facts that can be obtained from such data. These exclusions include information about what, how, when, where and why a particular piece of information was processed, encoded into data and stored during business operations or decisions making cycle. The contexts of business actions for the facts that are obtained from data that stored within the physical implementation of CSIS exist as part of the interactions of business rules that created such data. Thus, the facts represented by the data stored within database systems without the associated interactions of business rules may not have the same contexts of business actions as that of the data captured and encoded. In order to preserve the semantics of information presented to end-users through BI, design of CSIS need to encode facts with the contexts of business actions and store them as data with associated interactions of business rules. BI solutions can use these same business rules to decode the data stored within database systems into the facts with the contexts of business actions. Facts with the contexts of business actions enables end-users to ‘process information into knowledge’, which are then used to support business operations, collaborations and/or decision making. Design of CSIS needs to consider the capturing, encoding, recalling and decoding of data, whiles putting particular emphasis on maintaining the semantics of data captured. To achieve this would require an integrated approach to the design of domain models and data sources for CSIS and to enable BI across the organisation. This suggests the need for a unified approach to the collection, processing and encoding of information into data, the recalling decoding of data into the facts with the associated contexts of business actions, and the storage of data with associated interactions with business rules. Such a unified design approach must not only consider integration between knowledge and operations support as described by [11]; but also the integration of knowledge, business collaborations, decision making and BI requirements. Such integration will ensure consideration for the preservation of semantics (i.e. fact with the contexts of business actions) of data captured and encoded. This will thus ensure the design of CSIS give due consideration to the capture, processing, encoding and storage of data with associated interactions of business rules.



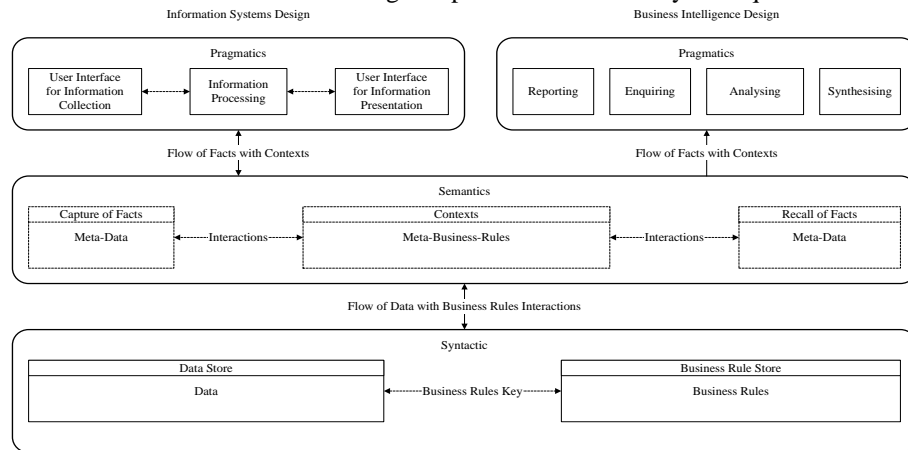
**Fig. 1.** Logical Design Pattern for Information Systems and Business Intelligence

### 3 Integration of BI into Information Systems Design

We propose LOPIBIS as a logical pattern for integrating BI into CSIS design. The proposed integration is to enable for the use of a single version of business rules (SVOBR) to capture, process and recall same sets of data. LOPIBIS follows the three principal layers of logical design that were described by [10] (i.e. presentation, domain model and data sources), except that LOPIBIS considers the interactions of business rules with data during the design of domain models and the storage of data with associated business rules interaction into respective information sources (see Fig. 2). The design pattern for the presentation layer of LOPIBIS remains the same as the current logical design pattern as described by [10]. That is, it is composed of two main information presentation design sets. One design set for the presentation of information to support business operations, collaboration and decision making (i.e. information collection, processing and display); and the other design set for the presentation of information to satisfy business intelligence requirements (i.e. reporting, inquiring, analysing and synthesising information). Information presentation design for BI considers the delivery of functionalities that enable capabilities for end-users to report, enquire, analyses and synthesises the facts with contexts of business action of data obtained from multiple and disparate business information sources. LOPIBIS integrates the domain model design of into consistent and unified facts with contexts of business actions (CUFCOBA). The CUFCOBA model contains two sets of meta-data and one set of meta-business-rules. One set of meta-data describes the business data capture, while the other describes business data recall. The meta-business-rules describe the business rules interactions that process, encode and decode business data. One set of meta-data are designed with associated meta-business-rules to capture facts with contexts of business actions and encode it into data for storage. The other set of meta-data are designed with associated meta-business-rules to recall and decode data into the facts with contexts of business actions for presentation. These two sets of meta-data and the meta-business-rules within the CUFCOBA model are used to provide support for the storage and sourcing of data with business rules, which include the extraction of data with business rules interactions either during processing of information or integration of information from disparate sources.

McFadden and Hoffer [12] quoted in [13] describe meta-data as “data about data”. Meta-business-rules, however, represent business rules about the interactions of business rules. Meta-data and meta-business-rules together provide information about why, when, who, where, when, what and how a particular piece of data is captured, processed, stored and recalled. The CUFCOBA model ensures the design of meta-data and meta-business-rules use a locking and key mechanism with a business rules key and associated business rules executions key attached to each piece of raw data. Implementation and use of systems that are based on the CUFCOBA logical design pattern will help to eliminate the use of MVOBR within physical implementations of CSIS and BI. Therefore, CSIS that are designed based on the LOPIBIS, will bridge the semantic gaps between the facts with contexts of business action of data captured and stored within physical implementations of CSIS and the facts with contexts of business action that are obtained from the data recalled and presented through BI.

Semantics contents (i.e. facts with business activity context) are encoded into data with business rules for computers to store using suitable database management systems and business rules management systems. LOPIBIS integrates information source design of CSIS and BI into a unified data with business rules (UDBR). UDBR contains a data store and a business rules store; it does focus design attention on the storage of data with business rules interactions, rather than the storage of just data. Data with business rules interactions stores are designed to be implemented using data with business rules interactions management systems (DBRIMS), while associated business rules stores are designed to be implemented using business rules management systems (BRMS). An Implementation of CSIS that is based on CUFCOBA and UDBR logical design patterns would enable CSIS to capture and recall data and BI to recall the same set of data with the same set of business rules. Facts with context of business actions are implemented using a set meta-data and meta-business-rules. These sets of meta-data and meta-business-rules are used to encode and store facts with contexts of business actions into data with business rules interactions; and also to source data with business rules interactions for decoding into the facts with contexts of business actions. This ensures the semantics contained business data that are captured and stored using the physical implementations of CSIS are maintained when such data are recalled into for decoding and presentation to satisfy BI requirements.



**Fig. 2.** A Logical Pattern for Integrating Business Intelligence into Information Systems Design

### 3.1 Integrating BI into Semantic Models

The integrated design of semantic layers with considerations for BI requirements would allow for the implementation and use of two sets of meta-data and a set of meta-business-rules, which are used to:

- Capture facts with the context of business actions and encode them into data for processing and storage along with the interactions of business rules.

- Recall and decode data with associated business rules interactions into facts with the context of business actions that are presented for purposeful use.

Meta-data with associated meta-business-rules for data capture describes how facts with the context of business actions are captured and encoded into data with the interactions of business rules, and how to store these into respective data or business rules stores. The meta-data with associated meta-business-rules that are used for data capture should be designed as a locking mechanism with an informative collection key (ICK). ICK will be used by application programs to encode captured facts and the context of business actions into data with the interactions of business rules, which can then be stored into their respective data and business rules stores as business rules key. Meta-data with associated meta-business-rules for data recall, describes how to recall and data with business rules interactions, and how to decode them into facts with the context of business actions. The meta-data with associated meta-business-rules that are used for data recall should be designed using an unlocking mechanism with a business intelligence key (BIK). BIK will be used by application programs to recall and decode data with the business rules keys that are associated with the original interactions of business rules into facts with the context of business actions. For each ICK within the unified semantic model, there must be a corresponding BIK. This ensures the semantics of data captured are maintained when the data is recalled, irrespective of the original purpose of the design of the application program that was used to capture the data. An integrated approach to the design of semantic models for implementation of application programs that are used to provide support for business operations, collaborations, decision making and BI; would enable organisations to implement and use consistent facts with the contexts of business actions within and across business domains. Such semantic model will have four main functions, which are to:

- Facilitate for the collection, processing and display information to provide support purposeful use of business operations, collaborations and decision making; and to provide capabilities for reporting, inquiring, analyzing and synthesizing of information as dictated by BI requirements.
- Encode facts with the contexts of business actions into the data with the interactions of business rules for storage into a unified organization-wide information store.
- Decode data with the interactions of business rules that are sourced from the unified organization-wide information store into the facts with the contexts of business actions.
- Facilitate for storage of data with the interactions of business rules that are used to capture, encode, process, recall and decode the data into the unified organisation-wide information store.

The integrated semantics model meets the following criteria, which we argued are needed to help resolve semantic gaps between CSIS that are designed to provide support for business operations, collaborations or decision making and BI:



- CSIS are designed to consider the display of information, processing with business rules and storage of data with the interaction of business rules that generated the data.
- CSIS are designed to recall data and decode them into facts with context based on data with consideration to business rules interactions.

### **3.2 Integrating BI into Syntactic Models**

The CUFCOBA semantics, design pattern enables the design and implementation of domain models that are based on common sets of meta-data for data capture and recall and meta-business-rules for their interactions with business rules. These meta-data and meta-business-rules can be used to recall data with associated business rules interactions for decoding into the facts with the contexts of business actions without the need to for a separate data storage design layer for business intelligence specific data. Therefore the need for ETL process becomes unnecessary, particularly the cleansing and transformation of data. Extraction of data with associated business rules interactions from one information source to another may be necessary for the purposes of information backup and consolidation, as well as, other business, functional or technical reasons. In recent years, technologies such as in-memory database architectures, disk-based architectures and many more have been made available to offer fast read-write to and from data stores or business rule stores. Also, technologies that enable the use of meta-information on data and business rules are available to enable separation of applications from data and business rule stores. Availability of these technologies makes it possible for the implementation and use of UDBR models. Such implementation in an organization will enable storage and sourcing of data with business rules and will invalidate the need for ETL, particularly the cleansing and transformation of data. Extraction of data with business rules from one UDBR source into another may be necessary for the purposes of backup, consolidation and other business, functional or technical reasons. BI can thus be used to recall and decode data with associated business rules interactions into the facts with the contexts of business actions, without the need for separate data storage for BI or the associated ETL process. These enable the facts with the contexts of business actions that may be presented through BI to be based on data with the associated business rules interactions that generated the data, instead of being based on data with recreated versions of the interactions with business rules. Thus, it ensures the semantics of the data captured, processed and encoded during business operations, collaboration and decision making are preserved in the data recalled and decoded during BI.

## **4 Discussions and Conclusion**

BI enables use of evidence from past business actions and decisions to support decisions and actions in the present. It also enables use of evidence past and present actions and decisions to anticipate or predict future business actions and decisions and

their possible effect on the organization and its environments. The current design of domain models and information sources for an information system gives little consideration to the requirements to enable BI. The requirements for BI are considered when end-user communities start to request for information from the physical implementations of CSIS to satisfy complex business reporting and analytical requirements. Usually reports produced from these physical implementations of CSIS to satisfy these requests are satisfactory for BI purposes when the volume, velocity and variety of information are very low. Thus, as the volume, velocity and variety of data captured by BSS increases, so does the BI obtain from these systems becomes limited. The disparate nature of CSIS that are implemented across an organisation, presents particular challenges when attempting to obtain BI from these disparate CSIS without the original business rules used to capture, process and encode the data. New sets of business rules that may not be the same as the original business rules are used within BI to decode the data into the facts with contexts of business actions. Without the original business rules and associated interactions, the facts obtained from data stored lacks the contexts of business actions; and facts without the contexts of business actions encourages the interpretation of facts with made-up contexts of business actions or to put it simply, misinterpretations. The consequences of these include the:

- Problems associated with inconsistent versions of the same information that may exist in an organization. These are usually due to the use of MVOBR to capture, process and recall from multiple and often disparate data sources.
- Gaps in the semantics between data captured and stored within physical implementations of CSIS and data recalled and decoded into information through BI. The gap is worsened by lack of contextual details about the circumstance in which the data was captured or simply put, lack of data with its original contexts.

Various degrees of differences exist in the interactions of business rules through which information is collected, processed and encoded into data and the interactions of business rules that recall and decode data into the facts with contexts of business actions for presentation through BI. These differences in the interactions of business rules can partly be attributed to the lack of consideration for BI requirements during CSIS design. As a result, most implementations of CSIS would have business rules embedded into the very core (i.e. application code) of these systems. These make it very difficult to recall and decode data stored with the business rules interactions that generated the data; and therefore difficult to recall with the data, the context within which it was captured, processed and encoded. Especially when the data is being recalled from an information system that is not part of the original information system used to capture and encode it. It is, thus, often practical to extract only the data and then transform it into information using a new set of business rules. LOPIBIS is composed of separate pragmatics design layers for CSIS and BI, an integrated semantics design layer called CUFCOBA and an integrated syntactic design layer called UDBR. We argued the CUFCOBA design pattern is needed to help resolve semantic gaps between data that are captured, process and stored within the physical implementations of CSIS and data that are recalled and decoded within BI. The UDBR model advocates integration of BI into the syntactic design layers of CSIS. The UDBR de-

sign pattern enables implementation of information storage systems for data with business rules interactions by co-designing databases and business rule stores. This offers a unified approach to the capturing, processing, encoding, recalling and decoding of data both in the physical implementations of CSIS and BI; ensuring the facts obtained from data that are recalled through BI will have the same contexts of business actions as that of the data that were captured, processed, encoded and stored. Thereby helping to preserve the contexts of business actions of data regardless of the purpose and mechanism that is used to recall and decode the data.

## References

1. L. Duan and L. Xu, "Business Intelligence for Enterprise Systems: A Survey," *Industrial Informatics, IEEE Transactions on*, vol. PP, pp. 1-1, 2012.
2. D. Remenyi and M. Sherwood-Smith, "Business benefits from information systems through an active benefits realisation programme," *International Journal of Project Management*, vol. 16, pp. 81-98, 1998.
3. R. C. Basole, *et al.*, "Visual Analytics for Converging-Business-Ecosystem Intelligence," *Computer Graphics and Applications, IEEE*, vol. 32, pp. 92-96, 2012.
4. J. Shim, *et al.*, "Past, present, and future of decision support technology," *Decision support systems*, vol. 33, pp. 111-126, 2002.
5. R. P. Mosimann, *et al.*, *The Performance Manager : Proven Strategies for Turning Information into Higher Business Performance*. Ottawa, Ontario, Canada: Cognos Press, 2007.
6. G. Ariav and M. J. Ginzberg, "DSS design: a systemic view of decision support," *Commun. ACM*, vol. 28, pp. 1045-1052, 1985.
7. B. Wixom and H. Watson, "The BI-based organization," *International Journal of Business Intelligence Research*, vol. 1, pp. 13-28, 2010.
8. L. T. Moss and S. Atre, *Business intelligence roadmap: the complete project lifecycle for decision-support applications*: Addison-Wesley Professional, 2003.
9. C. Fisher and B. Downes, "Performance measurement and metric manipulation in the public sector," *Business Ethics: A European Review*, vol. 17, pp. 245-258, 2008.
10. M. Fowler, *Patterns of enterprise application architecture*: Addison-Wesley Professional, 2003.
11. W. F. Cody, *et al.*, "The integration of business intelligence and knowledge management," *IBM Systems Journal*, vol. 41, pp. 697-713, 2002.
12. F. R. McFadden and J. A. Hoffer, *Data Base Management*. Menlo Park, CA: Benjamin-Cummings Publishing Co., Inc., 1988.
13. J. A. Ricketts, *et al.*, "Data reengineering for application systems," in *Software Maintenance, 1989., Proceedings., Conference on*, 1989, pp. 174-179.