OO Enterprise Architecture approach using UML :Information System Architecture Perspective

James Leo

Principal Consultant
Object Oriented Pty Ltd
Email: jleo@oopl.com.au

ABSTRACT

Globalisation of the economy, aggressive competition and the widespread use of new technology, Internet and e-commerce requires new thinking in the way business is operating. This challenges lead to urgent needs and greater reliance on Information Technology (IT) solutions.

Software architecture has been recognised as the development product that gives the highest return on investment with respect to quality, schedule and cost. The emergence of Object Oriented (OO) approach as applied to large scale enterprise software solutions requires closer examination on the impact and adequacy of OO paradigm approach and the use of Unified Modelling Language (UML) in meeting these enterprise architecture challenges.

This paper looks at the Object Oriented Enterprise Architecture approach using UML as informed by the Information System Architecture (ISA) Framework perspective.

1. Introduction

Information Technology (IT) has been recognised to play a critical enabling technology role for many organisation embarking on business re-engineering process [5]. Many organisations and corporations now recognise that achievement of the desired business strategic outcomes is closely linked to the effectiveness and efficiency in using IT.

This paper looks at the Object Oriented (OO) Enterprise Architecture approach using Unified Modelling Language (UML) as informed by the Information System Architecture (ISA) Framework perspective [7, 10].

1.1. Background

In the 1990's most corporations found themselves operating in an environment characterised by rapid change and uncertainty about the future. Globalisation of the economy, aggressive competition and the widespread use of new technology requires new thinking in the way business is operating and the business processes that are used to produce the goods and services. The changes wrought by e-commerce, Internet and Web technology have initiated a paradigm shift in business and consumer's expectation and behaviour. [2, 8].

However, there has been very little guidance available to help organisation reaping the benefits promised by the use of IT solutions. The sad fact is that we now have a situation whereby one year of business re-engineering can result in typically 4 years of IT development and implementation [9]. The rapid pace of technology and changes in business environment almost guarantee that corporation use of IT is always lagging far behind their business needs.

Software architecture has been recognised as the development product that gives the highest return on investment with respect to quality, schedule and cost [1]. The conceptual integrity of software architecture has been argued to be essential in developing good software system or product [3]. The emergence of Object Oriented approach as applied to large scale enterprise software solutions requires closer examination on the impact and adequacy of OO paradigm approach and the use of UML in meeting these enterprise architecture challenges.

2. ISA Framework – An Overview

ISA Framework provides a logical structure for classifying descriptive representation of an enterprise model and its associated planning, design and implementation artefacts and deliverables. This framework is derived from a similar classification structures that are found within the older Architecture/Construction and Engineering/Manufacturing disciplines [7, 10].

ISA Framework has been frequently employed as a logical structure to describe the enterprise-level Information Engineering modelling needs for large companies. Its classification mechanism provides distinctive views of an enterprise model from the perspective of stakeholders along the line of "*Planner, Owner, Designer, Builder and Component/Representation*". Therefore it provides a structure to identify and describe different views from the various stakeholders involved in determine the scope, specification, analysis, design and implementation of a system or an enterprise wide solution.

The following table describes the view on the architectural representations produced over the process of building a complex engineering project, along with analogs in the building, airplane and information systems communities ([10] p 282, Table 2):

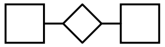
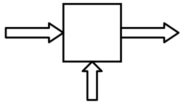
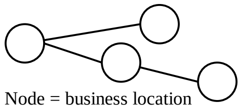
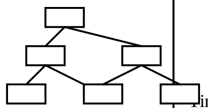
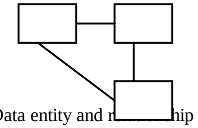
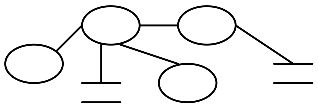
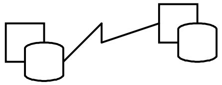
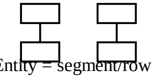
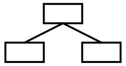
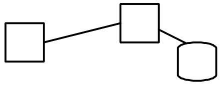
Generic	**Buildings**	**Airplanes**	**Information Systems**
Ballpark	Bubble charts	Concepts	Scope / objectives
Owner's representation	Architect's drawings	Work breakdown structure	Model of business / business description
Designer's representation	Architect's plans	Engineering design / bill-of-materials	Model of information system
Builder's representation	Contractor's plans	Manufacturing engineering design / bill-of-materials	Technology model
Out-of-context representation	Shop plans	Asembly / fabrication drawings	Detailed description
Machine language representation	----	Numerical code programs	Machine language / object code
Product	Building	Airplane	Information system

Table 1 Architecture Artifacts

The ISA Framework (see 2.1 ISA Framework Diagram) attempts to provide a view on the enterprise model's "*What, How, Where, Who, Where and Why*" characteristics and rationale along the perspective of the various stakeholders. This is based on the idea that stakeholders would view the system from their own unique perspective. This viewpoint is highly dependent on individual stakeholder's specific roles.

2.1. ISA Framework Diagram

The following diagram summarises the key aspect of ISA framework ([7], figure 6 p600-601).

	Data *What*	**Function** *How*	**Network** *Where*	**People** *Who*	**Time** *When*	**Motivation** *Why*
Scope *Planner*	List of things important to the business Entity = Class of Business thing	List of processes the business performs Function = Class of Business Processes	List of locations in which the business operates Node = Major business location	List of organisations/agents important to the business Agent = Major organisation unit	List of events significant to the business Time = Major business event	List of Business Goals and Strategy Ends/Means = Major business goal/critical success factor
Enterprise Model *Owner*	Semantic Model using Entity-Relation diagram 	Process Flow diagram 	Logistic network  Node = business location Link = business linkage	Organisation chart  Agent = organisation unit Work = work product	Master schedule Time = business event Cycle = business cycle	Business Plan Ends = business objective Means = business strategy
System Model *Designer*	Data Model  Data entity and relationship	Data Flow Diagram 	Distributed System Architecture  Node = IS function Link = line protocol	Human Interface Architecture Agent = user Work = deliverable	Processing Structure Time = system event Cycle = processing cycle	Knowledge Architecture Ends = criterion Means = option
Technology Model *Builder*	Data Design  Entity = segment/row Relationship = pointer/key	Structured Chart 	System Architecture  Node = hardware/system/software Link = line specification	Human/Technology Interface Eg. Presentation Architecture Agent = user Work = job	Control Structure Time = execute Cycle = component cycle	Knowledge Design Ends = condition Means = Action
Components *Artefacts*	Data Definition Description Entity = field Relationship = address	Function Program Function = language statement Control block	Network Architecture Node = address Link = protocol	Security Architecture Agent = identity Work = transaction	Timing Definition Time = interrupt Cycle = machine cycle	Knowledge Definition Ends = sub-condition Means = step

2.2. ISA Framework Classification Rule

The following rules are used to classify the various enterprise model's analysis, design and implementation representation into the ISA Framework [10]:

Rule 1. The columns have no order.
There is no natural order, sequence or priority among all the model abstractions in various rows or columns. No one column is more important than the other is.

Rule 2. Each column has a simple, basic model.
Each column represents a generic metamodel abstraction that meets the "*what, how, where, who, when, and why*" perspective of the enterprise.

Rule 3. The basic model of each column must be unique.
Each column generic metamodel abstraction is unique as it serves to meet the separate concepts contained within the "*what, how, where, who, when, and why*" perspective of the enterprise.

Rule 4. Each row represents a distinct, unique perspective.
Each row contains a metamodel abstraction that meets the needs of stakeholders such as "*Planner, Owner, Designer, Builder and Component/Representation*". Therefore, each row provides an enterprise model view that is specific tailored to its stakeholder's perspective.

Rule 5. Each cell is unique.
No metamodel entity can be found in more than one cell. This is because each column contains a unique basic model and each row has a distinct and unique perspective.

Rule 6. The composite or integration of all cell models in one row constitutes a complete model from the perspective of that row.

Rule 7. The logic is recursive and generic.
The framework logic can be used to describe anything that has an owner, designer, and builder who make use of material, function and geometry.

2.3. ISA Framework - Discussion Points

From section 2.1 ISA Framework Diagram above, we can see that a set of different representation and modelling techniques are being used to capture information and translate into knowledge required in producing the Enterprise Architecture solution. The ISA Framework above identified the use of many traditional structured analysis/design methodology and tool-set.

The views provided by the ISA Framework could facilitate the various stakeholders to strike a balance between holistic, contextual and pragmatic/implementation view on a system from the enterprise model perspective. This is because the ISA Framework:

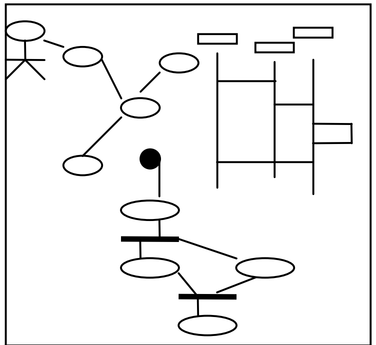
a) Provides a good classification scheme and abstraction on various enterprise model representations. This helps in improving understanding and communication among the various stakeholders;
b) Focuses on the variables applicable at each cell model;
c) Promotes awareness of contextual relationship that exist between the various enterprises model representations and artefacts. This helps to preserve the integrity of the enterprise model. It shall then helps to ensure that the system implementation meets the vision and needs expressed in the planning scope and analysis model.

3. OO Enterprise Architecture – ISA Framework Perspective

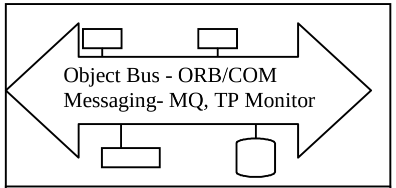
The following section 3.1 OO Enterprise Architecture using UML - ISA Framework Perspective diagram represented an attempt to apply ISA Framework concept in classifying and organising the descriptive representations of an OO Enterprise Architecture model approach. This work is derived from my current and past experiences in formulating enterprise and system level architecture models for several large organisations.

3.1. OO Enterprise Architecture using UML - ISA Framework Perspective

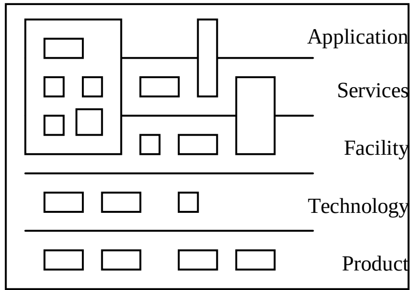
	Data *What*	**Function** *How*	**Network** *Where*	**People** *Who*	**Time** *When*	**Motivation** *Why*
Scope *Planner*	Enterprise data required to support Business Processes as in Business Process Map List of Business Objects (BO) for business requirements and data for specific system BO = Class of Business thing	Business functional requirements in Business Process Definition and Transaction Definition List of business functional requirements;	List of locations in which the business operates as identified in Business Process Map	List of Actors in Stakeholder Definition for enterprise as in Business Process Map List of Actors for a specific system in Use Cases	List of major business events in Business Process Definition and Transation Definition as in Business Process Map	List of Business Goals and Strategy
Enterprise Model *Owner*	Enterprise Data modelling, technique and process. Typical cases for Semantic Model uses Entity-Relational modelling or specific data modelling analysis tool System data and attributes using OO Business Object modelling approach as in BDM	Business Process Definition; Business Process Map; Transaction Definition.. Typical cases using TQM or BPR System function and behaviour using OO Use-Cases approaches in UML	Non-Functional Requirement Specification; Enterprise Architecture Reference Model; System Enterprise Model and Architecture	Organisation Chart; Actor roles, responsibilities and hierarchy is specified within Stakeholder Definition of enterprise Business Process Map and in the Actor diagram and description of BSM for a specific system	The timing of business events, business cycles and schedule for specific business process is described in Business Process Map and Transaction Definition using TQM or BPR tools and technique The timing of business events, business cycles and schedule for specific system behaviour is described using UML Sequence Diagram, State Transition Diagram and Activity Diagram. They are contained in BSM and BDM as appropriate.	Business Planning, Business Process Engineering , technique and process



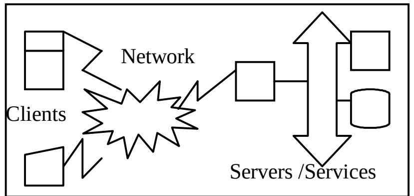
Business Function specified using UML
- Use Cases
- Sequence Diagrams
- Activity Diagrams



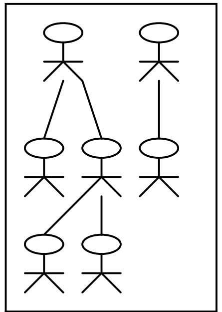
Objects Communication Architecture



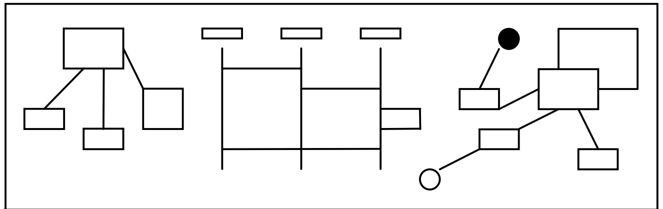
System Architecture Layers & Partitioning



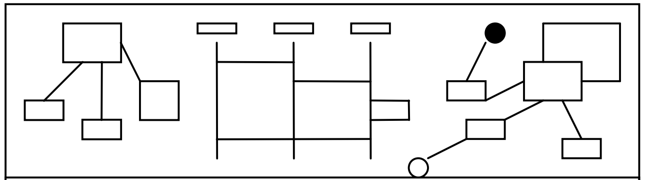
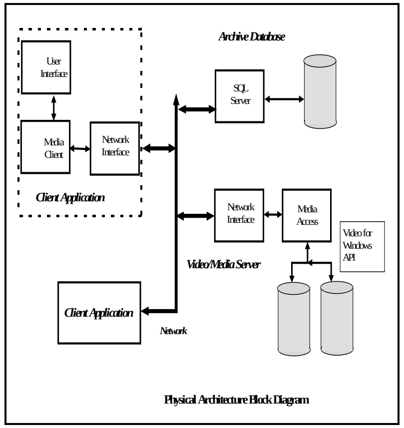
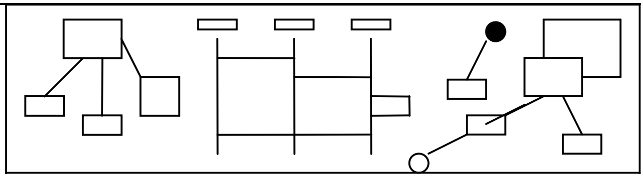
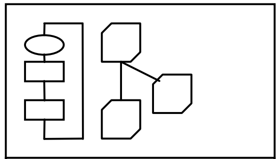
Logical System Architecture



Actor Diagram



Business Domain Model –BDM uses refined UML
- BO level Class Diagrams
- BO level Sequence Diagrams
- BO level State Transition Diagram

System Model *Designer*	 Solution Domain Model -SDM Include System Scenario Model - SSM ■ System level Class Diagrams ■ SSM Sequence Diagrams ■ State Transition Diagram Data design modelling, technique and process. Typical cases: Uses Logical Data Model or specific data modelling design tool		Architecture Solution  Physical Architecture Block Diagram	Actor interaction with system is specified using Use Cases Actor (person) or Actor (agent) is defined within design model Actor (agent) the Actor interaction with system is defined using System Use Case and System Sequence Diagram.	Any control structure, sequence and timing information is captured using Sequence Diagram and State Transition Diagram	Business Rules modelling, technique and process Typical cases: Business Rules are specified as part of Use Cases in BSM or as a constraints in the attributes and relationship of objects within the BDM
Technology Model *Builder*	 Component Model -CM ■ Class Diagrams ■ Sequence Diagrams ■ State Transition Diagram Data definition implementation, technique and process Typical cases: Manual or tool-based definition of database, tables, relations and constraints	Component Model addresses both Data and Functional level. It uses UML packages, class diagrams, sequence diagrams, STD as constrained by any specific technology adopted.	 Deployment Model Nodes Communication protocol and link Network transport Object communication mechanism	Actor (person) User Interface Design (not addressed by UML)	Control Structure and Timing technology technique and process. Typical cases: Event handling mechanism Trigger mechanism Store & Forward Work-flow mechanism OS or language specific features	Knowledge design technique and process Typical cases: Rules design is build into object behaviour, service and STM; Rules could also become conditions or constraints on object attributes, data structure or relationship; A rule-based data table could also be used to determine the object behaviour at run-time
Components *Artefacts*	Component / Class Specification describing attributes for persistence object classes; Data Definition - DDL scripts	Component / Class Specification describing packages, Class operations, Services and Interfaces; Program, scripts; DML scripts	Deployment Diagram ; Stylised Architecture diagrams; Network Communication CORBA; COM; RMI. Tool-based deployment script	Screens snapshots; Screen Prototype; Navigation Diagram; System Use Case; System Sequence Diagram; Actor Diagram; Organisation structure	Business Event; Timing for Events; Semaphore; MessageQueue; Job Control Language - JCL; Workflow;	Business strategy, Business Process; Business Rules, conditions and triggers; Domain Dictionary

Component Object – JavaBean; CORBA; OLE/ActiveX object	Extended by EJB; OTM; MTS
Interface Definition – IDL; MDL	Extended by ORB/DCOM/RMI

3.2. OO Approach Impacts on the ISA Framework Rules

The following are re-interpretation of ISA Framework rules (see section 2.2 ISA Framework Classification Rule) within the context of enterprise architecture work using object-oriented software engineering approach and technology:

~~Rule 2.~~ Each column has a simple basic model.
This is no longer upheld as development in Object Oriented software engineering has resulted in bridging the traditional chasm between data, function and network domain.

~~Rule 3~~. The basic model of each column must be unique.
This rule has been adapted to accommodate the Object Oriented paradigm. An object embodies both the data and function behaviour. Therefore, the use of object and object oriented programming language has resulted in the crossing the traditional chasm dividing Data and Functional model.
Similarly, CORBA, COM or Java has extended object beyond Data and Function model to empower object across heterogeneous network and environment.

~~Rule 5~~. Each cell is unique.
This is no longer upheld within Object Oriented paradigm. This is because a model element could exist in more than one cell.

3.3. OO Enterprise Architecture using ISA Framework Perspective – Discussion Points

(Note: The following discussion points serve as a preliminary points to raise awareness and discussion topics in the coming 2nd Australiasian Workshop on Software Architecture.)

The classification of OO Enterprise Architecture using UML as informed by ISA framework perspective highlighted the following insights:

1. OO approach in Domain Modelling is composed from simpler re-useable objects. This is in contrast with the traditional IS approach which favours decomposing the enterprise model into orthogonal data and functional views.
2. OO approach unifies data, function and network enterprise model elements. This is especially significant with the emerging use of enterprise level OO middleware such as CORBA, COM and EJB technology.
3. Beware of different granularity of "Objects" being used. Each model element at different level of ISA Framework serves very different purpose, level of abstraction nature and perspective needs. The current UML language provides no distinctive features to highlight these differences. One possible approach could be through the use of <<stereotype>> identification tag.
4. There is a need to define better OO Enterprise Architecture artefact, Enterprise Information model, business process, business rules, integration process, user interface, technique, guidelines and maturity criteria. The current UML syntax and semantic notation including Object Constraining Language (OCL) has been found severe lacking in feature and support for these works.
5. The use of UML has been found to be significantly bounded to the design/implementation system data and function level.
6. Additional process, techniques and guidelines opportunity exist in addressing other significant enterprise model elements which are not adequately addressed by current OO and UML approaches. Some of these enterprise architecture level issues have already been addressed by the use of Process MeNtOR [6] which is a world class component based OO methodology.

4. References

1. Bass, L., Clements, P. and Kazman, R. (1998). *Software Architecture in Practice*. Reading, Mass.: Addison-Wesley.
2. Brockman, J. (1996). *Digerati*. Orion Business Books.
3. Brook, F. (1995). The Mythical Man Month – Essays in Software Engineering, 20th edition. Reading, Mass.: Addison-Wesley.
4. Fowler, M. (1997), *UML Distilled*, Addison Wesley.
5. Hammer, M. and Champy, J. (1993). *Re-engineering the Corporation.* New York: Harper Business.
6. OOP/L (1999), *Process MeNtOR version 4*, Sydney: OOP/L

7. Sowa J.F. and Zachman J. A. (1992), “*Extending and Formalizing the Framework for Information System Architecture*” in *IBM System Journal* vol 31, no. 3, 1992. IBM Publication G321-5488.
8. Tapscott, D. (1995). *The Digital Economy,* New York: McGraw-Hill.
9. Trudzik, L. and Edwards J. (1996). “The Marriage of Business Process Reengineering and Object Technology”, in *Object World Conference Proceeding 1996*, volume 2, pp. 391-404.
10. Zachman J. A. (1987), “*A Framework for Information System Architecture*” in *IBM System Journal* vol 26, no. 3, 1987. IBM Publication G321-5298.