Architecting the Problem Space: an Architecture Framework-based Method for Definition of User Requirements

Paul Logan[1]*, Dr David Harvey[2]

[1]Empel Solutions Pty Ltd, Eltham, VIC, Australia

[2]Aerospace Concepts Pty Ltd, Fyshwick, ACT, Australia

*E-mail: pwlogan@gmail.com

ABSTRACT

Architectural frameworks and model-based systems engineering have developed over recent years as effective methods for analysis of requirements and development of conceptual system solutions. The process usually starts with some form of statement of need from which system requirements are developed. Little attention has been placed on structured methods for content and form of the user needs document, that is, definition of the problem space.

The method described here combines architecture framework principles and model based systems engineering techniques to develop a coherent and consistent architecture description that identifies system users and other stakeholders, their business/operational roles, relationships, interactions, activities, and their needs of the system of interest.

The essential attributes of the problem space that must be addressed by a system solution are captured in a traceable and verifiable data model which can be reported as a set of concise and consistent information artefacts.

KEYWORDS: Architecture; framework; need; requirement; model.

1. INTRODUCTION

Effective system development commences with identification of user requirements - stakeholder objectives, goals and needs - for the system under development [1]. These user requirements address both immediate end-user needs and the broader business requirements of the enterprise or organisation undertaking the system development.

A significant body of knowledge incorporating structured analysis and later object orientated theory and practice has been developed over several decades to provide current practitioners with an ample quantity of process and method descriptions for *what* to do throughout the system development life cycle [1]. However, when addressing capability that is required of the system, the emphasis tends to be on *system requirements* (*i.e.* what the system has to do) rather than on what the user seeks to achieve when using the system of interest, as expressed by the *user needs.*

More recent thinking and practice have focused on the *enterprise architecture* paradigm in which system requirements and solutions (*i.e. system architectures*) are aligned demonstrably with the strategic aims and purposes of the organisation [2]. In this paradigm, analysis effort is focused on definition of *outcomes* (*i.e.* what is achieved by use of assets) rather than *inputs*, (*i.e.* what assets the organisation has or resources it consumes). This approach provides a *capability* focus - what can the organisation 'can do', rather than what the organisation 'is'.

While this architectural approach emphasises alignment of strategic goals, capability needs, sys-

tem requirements and ultimately systems solutions, both functional and physical, few of the architecture frameworks provide definitive architecture description development processes and methods, and deliberately so. This stance is consistent with the 'outcomes' rather than 'means' emphasis inherent in the enterprise architecture paradigm but it does little to foster effective and efficient architecture data gathering and description.

The lack of guidance and direction is most apparent in the critical activity of user goals, objectives and needs definition. Nonetheless a well-defined architectural framework, such as the United States Department of Defense Architecture Framework (DoDAF) [3], through definition of a comprehensive data model that includes user identification, business process definition, and operational item flows, can form a robust basis for user needs elicitation and description.

The method for user needs definition described here incorporates model-based systems engineering principles and architecture description concepts. It involves the development of an information model, the starting point of which is the identification of business goals and objectives. It includes identification of stakeholders, their operational or business roles and the activities performed by stakeholders and the interactions between stakeholders necessary to achieve the operational tasks and the overall mission or goal. When the information model is sufficiently complete – as determined by a 'fitness for purpose' assessment – the data is analysed to identify user needs of the system of interest.

The needs, expressed as *user requirement statements*, both functional and non-functional, are also incorporated into the information model. They are demonstrably traceable to a class of user, the activities performed by the user class that generates the need, and the context in which the need occurs.

All of the data contained in the model can be reported in the form of architectural views constituting an architectural description. The specific format and content of any given view is determined by the analyst's choice or a mandated architecture framework definition or standard. The method as described here uses the DoDAF operational views as a framework for user needs analysis and description. Alternative well-defined frameworks could also be used [2,5,6].

2. ARCHITECTURE DESCRIPTION

Architecture is defined as the fundamental organisation of a system embodied in its components, their relationships to each other and to the environment and the principles guiding its design and evolution [4]. As depicted in Fig. 1 (redrawn and simplified from [4]), an architecture description includes a number of *views* that address stakeholders concerns. A view is a representation of a related set of architectural data using formats or models and a *model* in architecture description usage is a template for presenting architectural data.

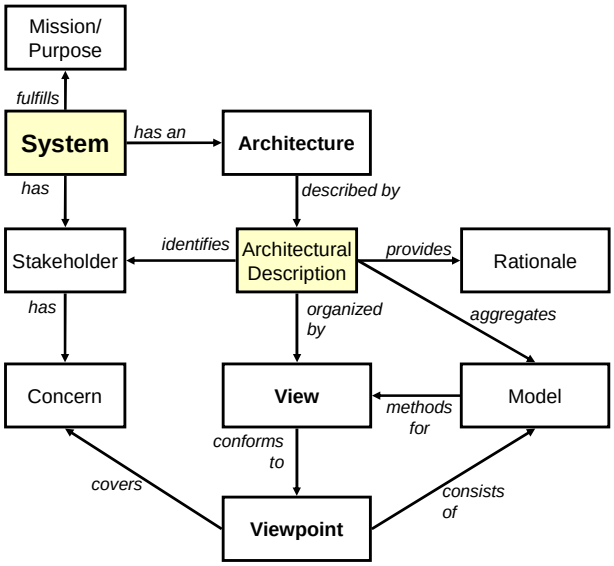

Fig. 1. Architecture description elements

Stakeholder concerns are identified from the perspective (or *viewpoint*) of a particular stakeholder class (*e.g.* operations, engineering, finance, support etc). The scope of the data required to address a particular stakeholder viewpoint is determined by the nature of the stakeholder concerns. Stakeholders may have very specific point issues such as circular probability error of missile impact or very broad issues such as through-life cost of ownership. Consequently, the views within a viewpoint may be few in number with limited content or be numerous and data-rich.

Architecture framework definitions have been developed to standardise the viewpoints, the views within those viewpoints, the content and in some cases the format of the view artefact. This is done to facilitate exchange and interpretation of architecture descriptions so that systems, both existing and planned, can be evaluated for such things as overall capability, cost of ownership, interoperabil-

ity and operational flexibility and adaptability.

3. NEEDS ANALYSIS FRAMEWORK

The DoDAF, in a manner similar to other frameworks such The Open Group Architecture Framework (TOGAF) [2], Zachman [5], and related military/defence architecture frameworks (such as [6]) identifies a number of viewpoints and views that can be used to describe system architecture. DoDAF was initially limited to four viewpoints: *operational*, *system*, *technical* and *all*, the later presenting general data relevant to the whole architecture description. Following developments of similar military capability orientated frameworks, and consistent with the enterprise architecture concepts of commercial business entities and non-defence government departments and agencies, the DoDAF now includes eight viewpoints [3]. These changes consist of three additional viewpoints – *capability*, *project*, *services*; the rearrangement of views to form a *data and information* viewpoint; and the renaming of technical as the *standards* viewpoint.

While relatively comprehensive if completed in detail, the DoDAF viewpoints and views are not the only viewpoints and associated views that can be used to describe a system and its architecture. A comprehensive model of a system is likely to include more data than is necessary to populate the DoDAF views as currently defined. For example, an emerging viewpoint, not formalized but of increasing significance, is the *human* viewpoint [7].

The key criterion for the content and level of detail associated with system architecture description is *fitness for purpose* [3]. The architecture description should capture and present data that is both relevant to, and at a level of detail appropriate to, the purpose of the architecture description. Relevancy and level of information abstraction are matters of analytic judgment, a benchmark being the level of assumed risk in relying on the data in the system model as the foundation for future work. A key test of fitness for purpose at any stage of model development is the extent of satisfaction of stakeholders concerns – the *raison d'être* of an architecture description.

The portion of the DoDAF of interest here is the *operational* viewpoint. The Operational Viewpoint (OV) captures the organisations, tasks, or activities performed together with the operational items that must be exchanged between them to accomplish the mission (or business purpose). The views (identified as models i.e. templates until populated with data) defined for the DoDAF operational viewpoint are listed in Table 1.

Table 1. Operational Views

ID	Name	Description
OV-1	High Level Operational Concept Graphic	The high-level graphical/textual description of the operational concept.
OV-2	Operational Resource Flow Description	A description of the resource flows exchanged between operational activities.
OV-3	Operational Resource Flow Matrix	A description of the resources exchanged and the relevant attributes of the exchanges.
OV-4	Organisational Relationships Chart	The organisational context, role or other relationships among organisations.
OV-5a	Operational Activity Decomposition Tree	The capabilities and activities (operational activities) organized in a hierarchal structure.
OV-5b	Operational Activity Model	The context of capabilities and activities (operational activities) and their relationships among activities, inputs, and outputs;
OV-6a	Operational Rules Model	Business rules that constrain operations.
OV-6b	State Transition Description	Business process (activity) responses to events.
OV-6c	Event-Trace Description	Actions in a scenario or sequence of events.

The set of views identified in the DoDAF operational viewpoint provides a useful framework for analysis and presentation of the system capability problem space. The viewpoint definition provides a model of the needs analysis model that the method described here instantiates.

4. MODEL OF MODELS

Model-based analysis and design develops a comprehensive model of the system architecture so that all views of the system are integrated and consistent. The model is an information model of the system of interest that consists of data elements that represent system elements, both conceptual and physical; the relationships between those elements; the actions performed by the elements and the items that flow so that the actions can be completed and coordinated for the system to achieve its overall purpose. Qualitative and quantitative information about these elements is captured in data attributes associated with each of the elements.

The element-relationship-attribute model structure is the essence of relational database the-

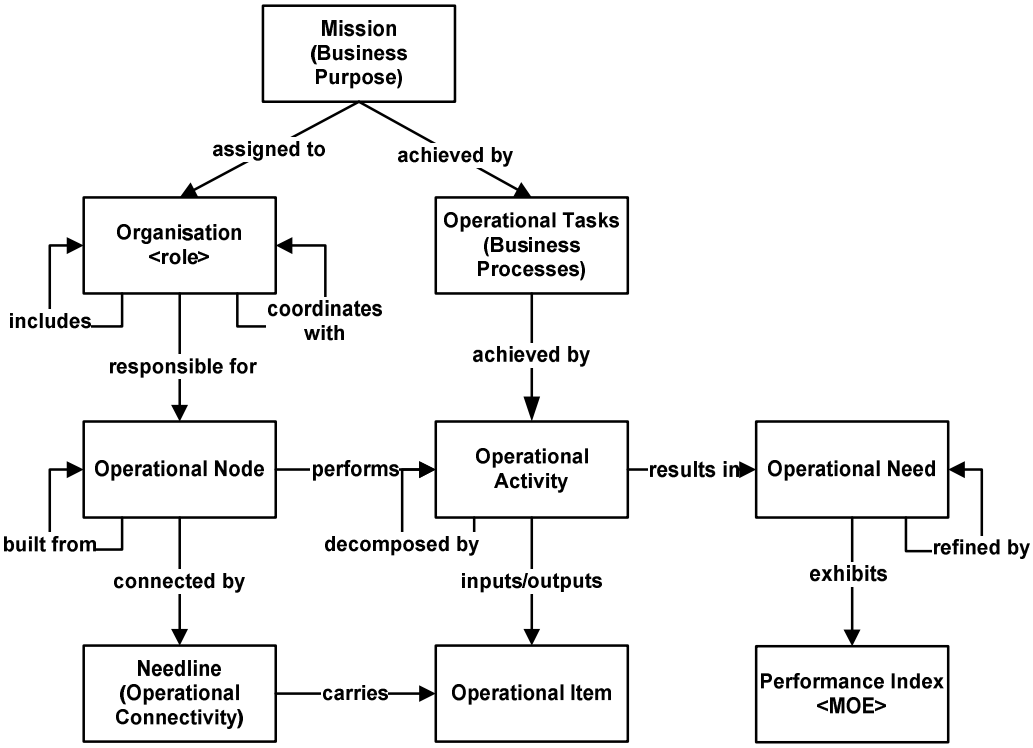

Fig. 2. Needs analysis meta-model (simplified)

ory and practice. Model-based systems engineering tools are applications built on relational database management systems [8]. The tool schema is a generic model or template, known as a *meta-model*, in the same way that the DoDAF viewpoint models are meta-models of the architecture views.

The meta-model for DoDAF's operational views and the reference model for the user needs definition method described here (shown in Fig. 2).

5. NEEDS ANALYSIS PROCESS

The process of populating the needs analysis model is simply that: populate the model, capturing data as it becomes available and refining the model at a particular level of abstraction until considered fit for purpose. A number of techniques can be used to identify and capture the data. Consistent with the enterprise architecture paradigm, an effective method is scenario- based needs analysis.

A scenario, as used here, is an assumed situation involving an organisational element tasked with achieving a self-selected or assigned mission or business purpose, and the context circumstances in which the goal is to be achieved. Once the scenario is documented, usually in high level terms and often graphically or pictorially, several analytic activities then follow (and not necessarily in the order listed):

- The mission is analysed to identify the operational tasks or business processes that contribute to achievement of the mission.
- All of the stakeholders or end-users involved in the scenario are identified together with their reporting and coordinating relationships.
- The stakeholders are categorised by type, *i.e.* class, and their operational role identified. Note that a single stakeholder class may have multiple roles, *i.e.* appears as more than one operating entity within the scenario.
- The activities that must be performed by the scenario participants, acting in their operational roles, to achieve the operational tasks and mission overall are captured. This is usually done via facilitated workshops in which stakeholders identify what actions occur, or should occur, in what sequence and with what interaction with other stakeholders for the tasks and mission to be achieved.
- When the model is completed to a satisfactory level of detail, through iterative application of the above analytic activities, the data captured is then analysed to identify the operational needs of each stakeholder (*i.e.* what the stakeholder must do to achieve or contribute to task and mission success) on the system of interest

and in particular the operational activities performed by the various stakeholders acting as their operational roles dictate.

No mandatory sequence of analytic activity is required for this model-based and scenario-based needs analysis method to be successfully applied. The activities are conducted in a sequence and manner appropriate to the overall purpose of the analysis. Colloquially, the method supports top down (conceptual "to-be" capability analysis and definition), bottom-up (existing "as-is" systems analysis, also known as "reverse-engineering") and middle-out ("a bit of both") development processes.

The "middle-out" process is the more common for most practitioners in that today most capability and system developments extend or improve existing capabilities and systems, rather than develop, *ab initio*, entirely new capabilities. The model-based approach facilitates working on that area of system definition that is most relevant or for which data is available and then using that portion of the model as a basis for elicitation or generation of additional system data.

6. METHOD APPLICATION

A process map that applies the method to a middle-out needs analysis is shown in Fig. 3. In this particular circumstance, an existing organisation is tasked with a mission that includes new tasks together with existing tasks the organisation is capable to achieving by using the existing system capability. The analysis task is to identify the needs that the organisation has of the system, which must be satisfied if it is to achieve the new tasks. The process in this situation starts with concurrent analysis of the mission and the organisation to which it assigned.

A high level operational concept is developed and captured as text or in graphic form. This is in essence a strategy or operational scenario for achieving the mission. It identifies, in broad terms, the participants, capabilities, resources and actions required to achieve the mission.

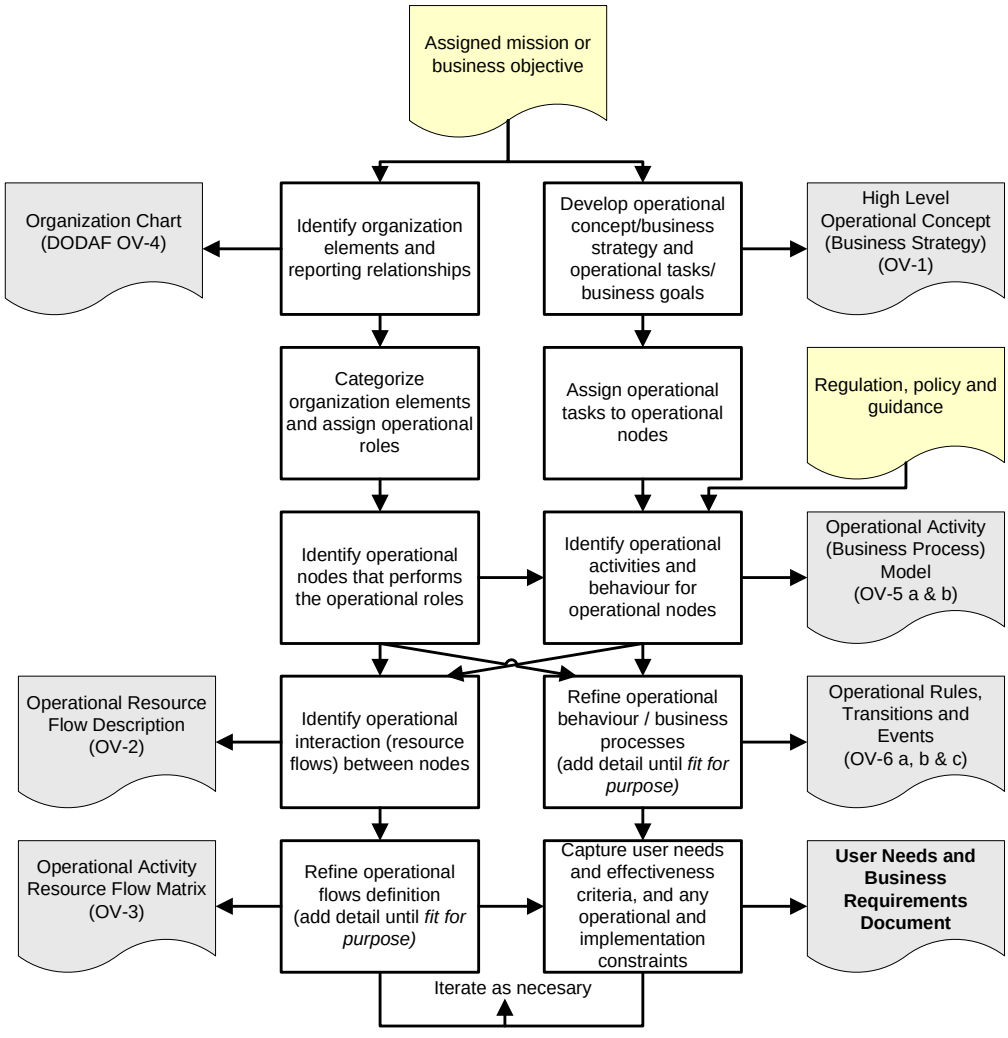

Fig. 3. Needs analysis process ("middle out" situation)

Mission analysis identifies all of the operational tasks that must be successfully completed by one or more of the stakeholders. Organisational analysis identifies all the stakeholders involved in the tasks and the relationships between the organisation elements (*i.e.* command and control lines). Mission and organisation analyses are conducted concurrently and iteratively as assessment of a task may identify the responsible organisation and *vice versa*.

The organisational elements are classified by type (*e.g.* business headquarters, data processing unit, tactical response group etc). The key information sought is the operational role of each stakeholder class element, because each role is represented in the subsequent operational model as an operational node performing assigned tasks. Each operational node then has or creates one or more operational needs – *i.e.* the node needs to perform some activity to achieve a particular outcome.

The architecture description product that results is an Organisation Chart, showing not only the formal reporting and coordination lines but also the role for each organisational element relevant to achievement of the assigned mission (see Fig. 4). Note that a single (real world) organisational element may have more than one operational role (usually performed at different times) – and a (conceptual) operational node is created in the model as the performing element for each role.

The tasks assigned to each operational node are decomposed into the necessary activities and resource exchange required for the tasks to be completed in accordance with the situation described in the scenario (see Fig. 5). The nodes may be either relatively free to act, or may be significantly constrained by statute, regulation, business policy and operational guidance. While in almost all cases some limitation on the freedom to act will be imposed on the stakeholders, the extent is usually related to the purpose for, and nature of the analysis: "as-is" analysis is constrained by current situation and circumstances whereas a new "to-be" capability analysis may initially be unconstrained in order to explore capability boundaries and options.

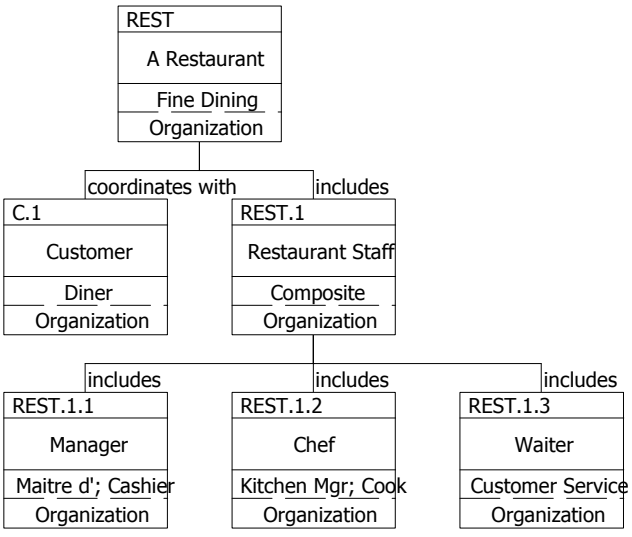

Fig. 5. Organisational relationships and attributes

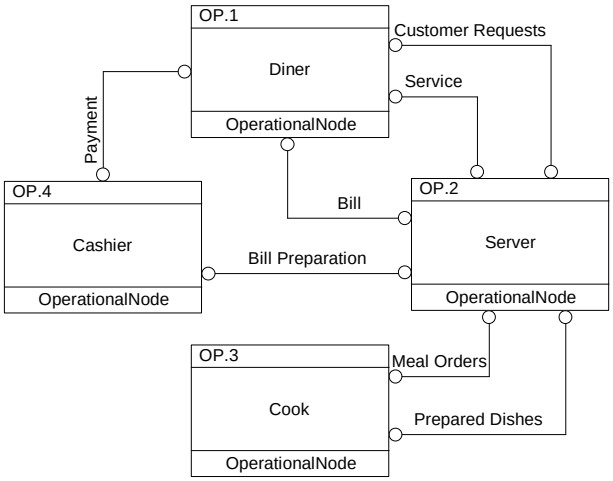

Fig. 4. Operational connectivity and flows

The effective capture and analysis of the operational activities performed by operational nodes is critical to accurate operational needs identification. A set of diagrams, or *constructs*, collectively called *behaviour diagrams* has been defined as part of the Systems Modelling Language (SysML) [9]. The behaviour diagrams, described in Table 2, can be used to capture and depict the behaviour of stakeholders *i.e.* capture their business processes and thus their needs of a particular system of interest. Note that in Table 2, the Use Case diagram is described as a composite construct making use of the other diagram constructs. The composite use case then becomes a scenario description of the type produced by application of the method described here.

Table 2. SysML behaviour constructs

Diagram	Diagram Purpose
Activity	Describes the flow of control and flow of inputs and outputs among actions
Sequence	Describes the message based sequence diagrams.
State Machine	Describes state based behaviour in terms of system states and their transitions.
Use Cases	Describes behaviour in terms of the high level functionality and uses of a system that are further specified in the other behavioural diagrams referred to above.

SysML templates and diagrams, based on the Unified Modelling Language (UML), a framework closely aligned with software definition and design, are too formalised or technical for most stakeholders to readily interpret. The products are highly suitable for use by practitioners within the system development community – they are less suitable for presentation to stakeholders at large for whom the systems are developed.

An alternative to the SysML and UML activity diagram is the Enhanced Functional Flow Block Diagram. (EFFBD) [10]. An EFFBD, as shown in Fig. 6, incorporates the functional/activity, data and control flows. This diagram format and diagramming technique are more readily comprehensible to end-users and other business level stakeholders and are preferred for the capture of user operational and business processes that are the basis of user needs derivation. The diagram, in fact a nested set of diagrams, captures the sequence, concurrency and alternative paths of all user activities that must be performed to achieve current or future operational tasks and business objectives derived during mission analysis.

All activities identified in the set of EFFBD are allocated to one or more of the operational nodes previously identified. Again, iteration between node and activity analyses is required to ensure completeness at a particular level for information abstraction. This allocation can be explicitly incorporated into the EFFBD by structuring the diagram into a number of concurrent behavioural threads each performed by one of the operational nodes. This is the *swim lane* technique commonly used in a number of diagramming notations.

The items that are produced or consumed by the activities are shown in the diagram as outputs from and inputs to the appropriate activities. Those items which exhibit control over the sequencing of activity performance are identified as *triggers* of the appropriate activity (shown with a double-headed arrow in the EFFBD). Each item is associated with the connectivity links between the nodes that perform the respective source and sink activities. At this stage, the key steps involved in developing the user needs analysis model are complete. The model is then analysed to derive the statements of user need.

7. DATA MODELLING - NOT DIAGRAMMING

It is important to maintain the emphasis on capturing data – entities, relationships and attributes. The essence of the model-based approach is the development of a data model of the system of

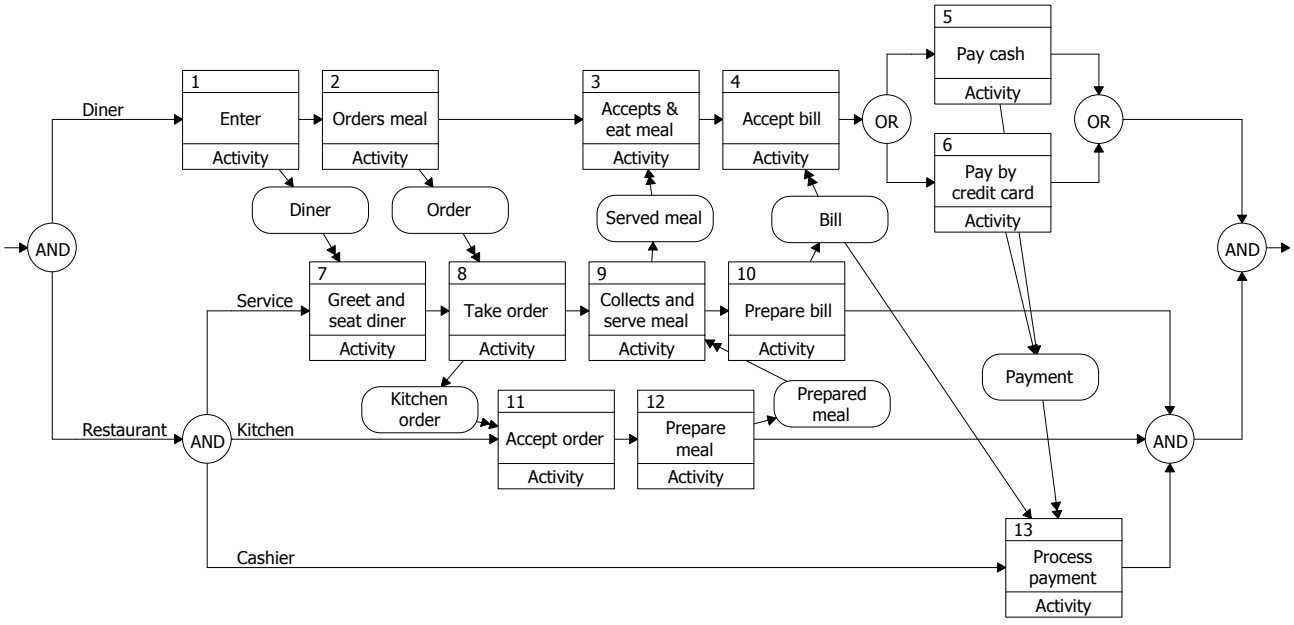

Fig. 6. Enhanced Functional Flow Block Diagram

interest. For user needs analysis, that system is the user task organisation or business entity.

As described in Section 2, the various diagrams and tables that are required as part of an architecture description or analysis report are only views on the data model. Consequently although a particular format for diagrams, especially that of the EFFBD, is preferred by the authors, the SysML suite of diagrams can readily be used to report the analysis and describe the operational architecture. This choice should be made by considering the intended audience and the complexity of the material to be presented.

8. USER NEEDS DERIVATION

Derivation of user needs is a relatively direct procedure once the data model of the uses operational architecture is complete. The user activities identified in the model are examined by asking the following question for each activity – *If that is what the user must do, then what does the user need of the system of interest to achieve a satisfactory outcome*?

As an example, in the EFFBD shown in Fig. 6, the waiter, operating as a service node, must meet and greet a dining customer. The waiter's user need of the Restaurant Operating System can then be stated as: *The waiter must be informed of a diner's arrival (within 30 seconds of the arrival*.)

Each activity performed by each node is examined in this manner and one or more needs statements derived. The collective needs can be published in the form of a User Requirements Document but should be maintained in the model, linked to the activity and node that generate the need. In this way a traceable needs exists within a structured framework consistent with the architecture description principles.

9. CONCLUSION

The operational (or business process) viewpoint of a rigorously defined architectural description framework provides a well-structured model for the often ill-structured analysis of user needs and capability definition. Combined with model-based analysis and reporting techniques, a process–independent method has been developed that provides rigour to capability systems analysis and user needs derivation without constraining the system stakeholders' operational concepts.

Use of the architecture framework operational viewpoint and views provides a robust and traceable basis for subsequent system requirements analysis and solution definition. The framework provides architecture (*i.e.* structure) for the *problem* space as well as the *solution* space.

REFERENCES

1. SE Handbook Working Group, *INCOSE Systems Engineering Handbook v3.2*, INCOSE, 2010.
2. The Open Group, *The Open Group Architecture Forum*, viewed 25 June 2010, <http://www.opengroup.org/architecture/>, 2010.
3. Department of Defense (DoD, USA) Chief Information Officer, *DoD Defence Architecture Framework (DoDAF) Version 2.0*, 2009.
4. IEEE 1471, *Recommended Practice for Architectural Description of Software-Intensive Systems Architecture Description of Software Intensives Systems*, 2000.
5. Zachman International Enterprise Architecture, *The Zachman Framework*, viewed 25 June 2010, <http://zachmaninternational.com>, 2010.
6. Ministry Of Defence (MOD, UK), *MOD Architecture Framework (MODAF) V1.2.004*, viewed 25 June 2010, <http://www.mod.uk/DefenceInternet/AboutDefence/WhatWeDo/InformationManagement/MODAF/>, 2010.
7. Handley, H. and Smillie, R., "Architecture Framework Human View: The NATO Approach," *Systems Engineering*, Vol. 13(1), pp. 72–79, 2007.
8. Estefan, J. A., *Survey of Model-Based Systems Engineering (MBSE) Methodologies*, INCOSE MBSE Focus Group, viewed 25 June 2010, <http://syseng.omg.org/MBSE_Methodology_Survey_RevA.pdf>, 2007.
9. The Object Management Group (OMG), *OMG Systems Modeling Language (SysML)*, viewed 25 June 2010, <http://www.omgsysml.org/>, 2010.
10. Long, J., *Relationships between common graphical representations in system engineering*, Vitech Corporation, 2002.