

The Role of the NGOSS Application Map

August 2005

Greg Fidler
Director,
AutoMagic KB LLC

Tel: +613 9557 2296
Mob: - +61 419 359 358
E-mail: - gregf@automagickb.com

Cliff Faurer
Director
AutoMagic KB LLC

Mob: - +1 (720) 323-4097
E-mail: - cfaurer@automagickb.com

ABSTRACT

This paper provides a definition of an application, and the role that the TM Forum Application Map can play within the context of an enterprise architecture reference model. In addition, it discusses the relationship of the Application Map to the key NGOSS artifacts, and to the various NGOSS views.



The TeleManagement Forum is developing an NGOSS Application Map whose purpose is described as follows¹:

The Business Problem

Within the TMF there is a strong definition of process (the eTOM) and data (the SID) but there lacks a formalised way of grouping together function and data into recognised components, which would then be regarded as potentially procurable as either applications or services

The project will produce an Application Map to provide this grouping - describing and naming a set of applications, together with the data they act upon/use and the function/process they perform and an element of decomposition/grouping. These applications are synonymous with services.

Value Contribution Statements

An Application Map will facilitate communication amongst BSS/OSS developers and users by providing a common language defined from a business view of the linkage between a process function and the data information model to the related application products or services that support the function. It will enable telecommunications personnel with a variety of skill competencies, not just developers, to identify and/or link process functions to their respective data information models and relate this linkage to individual BSS/OSS environments.

Customer Benefits

End user customers will be able to use an application map to communicate with Service Providers as they interact using customer initiated and controlled processes. Examples of end users are:

- SP when procuring a piece of software (end user buying an application)
- Internal SP organizations that need to interconnect with each other (both parties are end users as well as “application service provider”)
- Enterprise business subscribers to billing services from SP (end user buying a service)
- Vendors selling or hosting an application (like Siebel CRM on demand) (end user selling an application or service)

Service Provider Benefits

For Service Providers, an Application Map will serve as a vehicle for application portfolio management by providing a catalogue of systems, services and/or components that are described by using a common language. In addition, an Application Map will provide the following benefits:

- State procurement requirements in a way that can be easily understood by integrators and component suppliers.
- Define services independently of the technology or technologies implementing a service.
- Ensure that new systems can inter-work with existing ones.

Supplier Benefits

System Integrators along with Software and Hardware component providers will also benefit from using a common language with which to express component capabilities. It will enable them to easily transform the requirements that are received from service providers to deployable components that meet service provider expectations.

The Application Map needs to be developed in a way which re-enforces and builds on the developments of the NGOSS framework and architecture over the last few years. The

¹ TM Forum, Draft Application Map Charter v2.0, emailed to team for review, 21 June 2005

alternative, unless it is clearly positioned as an enhancement, is for the Application Map to be seen as competitive with the NGOSS framework, which is at odds with the TM Forums stated strategic objectives.

The charter captures many of the advantages correctly, but provides insufficient detail as to the relevant context in which the advantages may be achieved.

This white paper attempts to provide the necessary positioning of the Application Map as a standard reference model within the NGOSS framework.

Whilst not the major focus of this paper, the language within the charter tends to be focused on descriptions of functions and data. The NGOSS framework is structured assuming distributed information architectures, with a bias towards object-orientation. Therefore, the use of these words which tend to suggest the use of functional decomposition as an approach to architecture management can be confusing – even if not intended – as functional decomposition is, of course, a paradigm inconsistent with the NGOSS framework.

In order to avoid this possible confusion, all references to the value that applications deliver to a service provider in this paper are described using the more neutral words – business capability.

Definition of Application

Before the role of the Application Map can be described it is necessary to agree on the definition of “application”.

As part of considering what an application is, we will start by excluding those items that are not part of an application. The discussion is structured from the perspective of the typical view of the elements of a computing environment.

As a starting point it is assumed that an application is hosted on a computing platform which is not part of the application itself. This makes sense as a large UNIX server or an IBM mainframe can host multiple “applications” simultaneously, and in this space of “applications” the software development world has/is moving away from purpose built hardware. (Note that embedded applications are not being considered in this definition).

The next presumption is that an application uses or requires the services of an operating system. Once again I am making the general assumption that the application and operating system can be considered as separate entities (both logically and physically).

What is left? In broad terms, the things not yet considered are some form of database/datastore (which may or may not be required, which may or may not be separate to the software), a heap of middleware, communications infrastructure, test modules, etc. and lastly, some combination of software/software components purposely developed to achieve some business goal/objective. Except for the software/software components mentioned at the end, all of the other facilities (infrastructure and/or framework services)

mentioned have a common purpose independent of the type of business or organization using their services. Due to this, they are excluded from the definition of an application.

Since it would be possible to re-use software components combined differently to represent different business capability and therefore to achieve different business goals, it would not make sense to consider software components or some structure of software components to be the same thing as an “application”.

Given the above, it is clear that in SID terms an “application” has associations to Physical Resource, Logical Resource and Software, but is it possible to really touch and label an application? Is it “managed” in its own right or are its components actually being managed and the “state” of the “management application” is actually derived from the state of its individual components, all of which are managed entities? This later concept represents the “end of the line” of this reasoning. This means that in fact that an “application” is in reality a business concept (but a useful one) that has a management structure placed around it rather than being a managed entity in its own right. In a way it is a special type of management domain – an application domain.

Leveraging the definition of Management Domain from the SID, a preliminary definition of an application is as follows:

“An Application extends Management Domain (representing a special grouping of Managed Entities) and has two important properties.

First, it is used to partition managed objects into a meaningful logical grouping of business capability. One important use of this form of grouping is to allow for business decision-making in terms of investment, accountability, ownership, process alignment, etc. in relation to the managed objects. It also provides a means to show how business capabilities are distributed and scaled.

Second, it defines a common administrative domain that is used to administer the managed objects that it contains. This implies that all of the managed objects contained in this Application Domain are administered similarly - either by the same user, group of users or policy.”

Using this as the basis of a definition of an application, the role of the Application Map can be more clearly dealt with.

The Application Map can have several parallel legitimate uses. It can be used as a reference model, it can be used to categorize applications or it can be used to characterise applications.

Positioning the Application Map in the NGOSS Framework

The NGOSS Framework has been developed with the role of developing a Technology Neutral Architecture with a set of implementation neutral reference models (primarily the eTOM and the SID). However, while both the eTOM and the SID have high level conceptual views which can be used by management to represent aspects of the business, the primary purpose of both reference models is to provide a framework to be used by analysts and architects.

There is a need for an implementation neutral model which can be used to represent solution capabilities at a level where it is possible to provide a clear linkage between delivered business capability and management decisions based on investment, cost, operational accountability, etc. Neither the eTOM nor the SID, by themselves, provides an architectural view which can be used by management to meet these needs.

Application Map as a Categorization Model

At this point in time, there are very few Service Providers who are facing greenfield situations when it comes to deploying OSS and BSS solutions. Many Service Providers have been building and deploying applications of one form or another for many years. In addition various architectural approaches have been taken to designing these deployed applications.

In the older deployments, applications were often developed (by service providers or vendors) as a single monolithic entity with little thought given to architecting the developed application into a modularized overall solution architecture. The sorts of business capabilities that these applications deliver often have to be retrospectively assigned if a structured granular view of the business capabilities is required.

In later application development approaches, the internal application architecture has become increasingly more modular to decouple the application capabilities to allow for easier maintenance, enhancement and replacement. The current push towards “Service Oriented Architectures” (SOA) is taking this modularization to the next level of granularity. The underlying development techniques and programming languages have moved in parallel with this application architecture approach with the latest incarnation being based on object-oriented techniques and programming styles (and away from styles based on functional decomposition with associated data models).

In order for the Application Map to be useful as an industry reference model for formal **categorization**, the application categories must have enough span to allow the above broad breadth of application types to be categorized cleanly within the reference model. This means that the categories need to be fairly ‘wide’.

This role for the Application Map will allow for deployed applications to be grouped as necessary, and for appropriate management decision-making to be applied to the broad groups. However, it does have limitations in that within the broad groups different deployed applications will have more or less business capability.

Application Map as a Characterisation Model

The broad categorization use discussed above also has limitations when purchasing applications from software vendors.

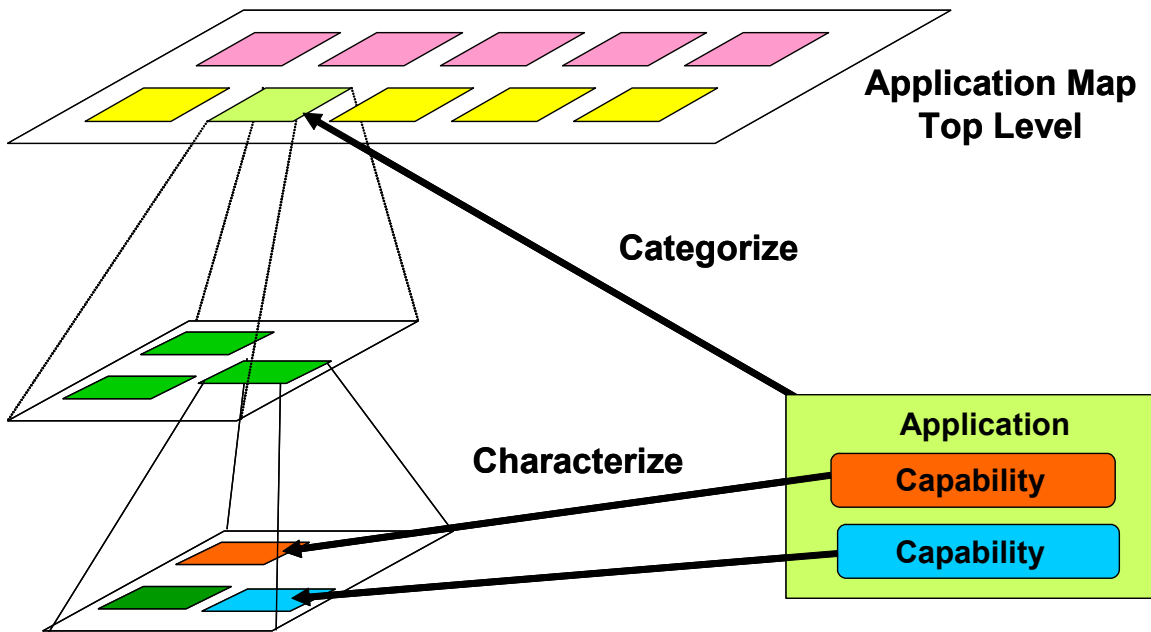
Software vendors, driven by their own commercial models, will develop their applications in accordance with their ability to develop applications to support their competitive advantage as they see it. By necessity this means that different software

vendors will package application capability differently, will describe their offerings differently, and will constantly re-position and enhance their applications based on changing market circumstances.

In this circumstance, the Application Map needs to have enough granularity to allow the service provider to accurately differentiate the capabilities delivered by applications from different vendors. The Application Map as a *characterisation reference model* needs to contain sufficient detail so that it can be used to clearly determine the low level capabilities provided by a vendor delivered application, and for that matter to clearly describe the capabilities of the already deployed applications. Without this, it will not be possible to manage application capability re-use, nor to make clear decisions as to whether available “off-the-shelf” applications can meet the business requirements.

Application Map as a Multi-Level Reference Framework

In order to meet both the categorization and the characterization requirements the Application Map needs to be developed as a multi-level framework, in much the same way as the eTOM and SID have been developed as a multi-level process framework and multi-level information framework, respectively. The figure below shows this pictorially:



The question naturally arises at this point – what is the lowest level of granularity required, or alternatively, what is the “bottom level detail” of the Application Map?

Whilst there is not a hard and fast answer to this question, the NGOSS architectural framework provides some clues.

The NGOSS architecture² must be a component-based architecture. From the minimalist point of view, components offer service capabilities. However, to fully understand the implications of component-based systems, the minimalist point of view is insufficient.

As a binary implementation of functionality, a component is an artefact. To be more specific, it is one of the artefacts of the software development process. In order to be useful, the services provided by a component must be clearly specified.

An NGOSS Component must:

- have all of its external (contextual) dependencies explicitly defined;
- be implemented in compliance to a technology-specific component model, such that a technology-specific component execution environment is capable of making the NGOSS Component services available at runtime to the remainder of the system
- either not have persistent state or must manage its persistency according to the business requirements of the system (e.g. high availability and/or hot failover). This applies to an NGOSS Component, not to an interface or service that will of course create, manipulate, and rely upon information that is stored in a persistent fashion within the system.

Given the defined services and the explicitly defined external (contextual) dependencies, an NGOSS Component can be combined in a system with other NGOSS Components to deliver an aggregated service that is then available to all other NGOSS Components (and client applications) within the system.

From the above definition, the lowest level of the Application Map would be represented by the components, which offer their services via defined interfaces, and which can be integrated by vendors to provide a different offering of application capability.

Application Map and the Information Model (SID and extensions)

This section considers the relationship between the Application Map and the NGOSS Information model (Shared Information & Data Model and extensions).

From the definition of application derived at the start, it is clear that an application can be considered as a holder of information and business logic. This information and logic jointly provides a business capability that is exposed through an interface. It is also clear that the same information and logic can be used in different combinations to provide a different, but possibly related, business capability. This means that the SID entities are not the lowest level of granularity, or the ultimate leaves, of the Application Map - they serve different purposes but are related.

Having said that, from a reference model perspective, whether particular types of application containers are being derived, at different levels of granularity, it is possible to establish a particular form of application as the “natural” home, in a containment sense, of particular information entities or groups of entities. As an example, a Resource Trouble Management application is very likely to be the ‘natural home’ of Resource Trouble Reports.

² TMF053 v4.0, December 2003, page11

Thus a mapping should be established between the elements of the Application Map and elements of the SID and extensions, at different levels of granularity.

Application Map and the Process Model (eTOM and extensions)

This section considers the relationship between the Application Map and the NGOSS Process model (the eTOM and extensions).

In the general definition of a process, there is the presumption that between the triggering and completion of a process step some interaction between the process logic and information occurs. Given that an application can be considered as a holder of information and business logic that is exposed through an interface, it is reasonable to assume that a particular process step would have an association with a type of application.

Thus a mapping can be established between the elements of the Application Map and elements of the eTOM and extensions, at different levels of granularity.

Application Map in the Business & System View

As part of discovering solutions to meet Service Provider business concerns, the NGOSS methodology involves undertaking a logical analysis of the business problem, first addressing the business requirements, then moving through to establishing the conceptual design within the System View.

Within the NGOSS technology neutral framework, three key reference models have associations with each other:

- Processes are supported by applications
- Processes interact with information
- Applications contain information

Therefore in the analysis of the business and system views, it is possible to use the associations between the three reference models, and the linkages between these reference models and actual process, application and data deployments, to quickly determine which aspects of the business problem may be solved by re-using available processes and capabilities.

Application Map in the Implementation View

Software vendors and Service Provider IT development teams are responsible for converting conceptual application designs into realized capabilities offered as deployable system solutions.

Increasingly, driven by economies of scale, Service Providers are looking to purchase applications “Off-The-Shelf” to meet their business automation needs. The problem for Service Providers is that Software Vendors develop their applications in accordance with their own competitive imperatives. The challenge in a competitive purchasing process is to ensure that what the software vendor is offering is clear, and that any comparisons made between vendor offerings are valid.

Historically, as Service Providers viewed all developed and deployed applications as potentially delivering competitive advantage, there was no common view amongst Service Providers as to how the totality of automated business capability should be “split” amongst applications. As a result, software vendors have had a free-reign in determining how to package business capabilities in their applications.

There are several outcomes as a result of this application development approach that need to be considered in terms of using the NGOSS Application Map:

- Each vendor has packaged and named individual application capabilities within the context of their own business model
- The marketing description of the vendor application capabilities and the actual capability as defined within the Application Map are likely to be at variance
- Vendor applications are packaged according to the vendor view of what makes sense to them, and not within a framework that exists at the “categorization” level of the Application Map.

The formal Application Map reference model provides a tool for removing the confusion and for characterizing the vendor application business capability against a single reference. This characterization of the deployed capability could provide the necessary insight and consistency to allow for assessment and evaluation of vendor applications in provisioning processes. The downside, of course, is that time and effort will be required by the vendors for the characterization to be undertaken and to be captured.

However, by capturing this information against the NGOSS Application Map, the Software Vendor is undertaking the characterization work in the knowledge that the output of the work can be used in a consistent way for multiple Service Providers across the wider marketplace.

Application Map in the Deployment View

As indicated earlier, Service Providers have been deploying applications for many years. The packaging of business capability by the Service Provider in the deployment of applications has been entirely driven by factors such as organizational ownership, development boundaries, previously installed legacy capabilities, and a host of other factors.

In addition, it used to be a given in application purchase during the telecommunications monopoly years up to the late 1980’s/early 1990’s, that it was acceptable to modify/enhance purchased applications to more clearly meet the stated business requirements. The purchasing challenge was to purchase an application that met as many of the needs as possible and then to customize to get the rest. The result was a ‘one-off’ application that was very expensive to maintain and upgrade.

For Service Providers that maintained IT development organisations, whilst some purchased capability was used, large amounts of business capability were directly developed with an eye on only meeting the needs of the particular service provider.

There are several outcomes from this deployment approach that need to be considered in terms of using the NGOSS Application Map:

- There is unlikely to be a one-one relationship between any business capability purchased from a vendor and any deployed capability in the Service Provider's environment
- There is unlikely to be a re-use of the name of the application purchased from the vendor within the Service Provider's environment
- The Service Provider is likely to use a name for the deployed application that is associated with its use, or the market it services, a name made up by the project team, or a name deliberately chosen with a view to "selling" the application to management.
- The business capability modularity of the deployed application, if any, was likely to be determined entirely by the development team that built it, or the vendor that sold it, or both.
- Due to a lack of clear reference architecture standards used by Service Providers in categorizing deployed applications, the deployed "application" may incorporate software, operating systems, hardware, robots, middleware, databases, etc. The use of the word "application" and what it means will vary between Service Providers, and even for deployments within a single Service Provider enterprise.

The formal Application Map reference model provides a tool for removing the confusion and for characterizing the available deployed business capability against a single reference standard. This characterization of the deployed capability will provide the necessary insight and consistency to allow for proper management of the deployed application base. The downside, of course, is that time and effort will be required for the characterization to be undertaken and to be captured.

By capturing application characterization against the NGOSS Application Map, and using the process and information models (eTOM and SID plus extensions), the Service Provider is undertaking this work in the knowledge that the output can be used in a consistent way to compare local deployed capabilities against purchasable capabilities – as there is some possibility that Service Providers as a group will be able to convince vendors to characterize their offerings against an industry reference framework rather than multiple local Service Provider reference frameworks.

Such an industry standard approach to application characterization will allow all involved stakeholders to gain new insight into not only the requirements of the Service Providers, but also the fuller capability offered by vendor applications. Improved visibility of the solution under discussion by all involved stakeholders will lead to reduced integration costs and improved interoperability resulting in overall time and cost savings and quicker time to market.