Modeling The Enterprise IT Infrastructure

An IT Service Management Approach

By: David Chiu
     D.L. Tsui

Version 1.2b

Acknowledgement

The authors would like to thank Troy DuMoulin of Pink Elephant Inc. for his contribution in the development of the original implementation object model. As a subject matter expert on ITIL, he provided a great wealth of knowledge on IT Service Management, which is the foundation of the object model design described in this paper.

Authors

David Chiu is an IT Service Manager at BMO Financial Group. He plays a key role in his company's ITIL Program as an internal ITIL Advisor and Process Manager. He is one of the process architects for many of the ITIL processes implemented in BMO over the last 5 years, including Release, Change, Configuration and Service Level Management. As a Process Manager, he provides leadership and subject matter expertise for the IT Service Management improvement initiatives undertaking by his organization. David has also been actively promoting ITIL best practices outside of his company. He has been a speaker in many of the IT Service Management conferences presenting topics on Configuration Management, COBIT, Service Management Tool selection and ITIL implementation. He won two awards (in year 2003 and 2004) for "Best ITIL Case Study" at the 7th and 8th Annual International IT Service Management Conferences hosted by Pink Elephant.

David has been working in the IT industry for more than 15 years, with business experience in the manufacturing and financial sectors. In addition to his IT business process engineering and continuous process improvement experiences, he has managed many IT projects and he has technical expertise in systems design, systems management and integration.

David has a Bachelor of Applied Science degree from University of Waterloo. He has obtained certification in ITIL and is currently working towards his PMP designation.

Contact: david.chiu@bmo.com

D.L. Tsui is a Senior Business Analyst at BMO Financial Group. During the last three years, he has been involved in the design and implementation of the ITIL Process Improvement Initiative at BMO (Project of the Year 2003). The process areas include Incident, Problem, Change, Configuration and Service Level Management. As an industrial engineer, he has also consulted for the Hong Kong International Airport Authority, Cathay Pacific Airways and Toyota Motor Corporation.

Contact: dunlop.tsui@bmo.com

Notice

The concepts and views expressed by the authors of this paper do not necessarily represent those of their employer, BMO Financial Group.

Table of Contents

1.0 Introduction

Organizations are increasingly recognizing ITIL as the standard for IT Service Management. ITIL provides a best practice process-based framework for supporting and delivering IT services to enable the enterprise to meet its business goals.

Central to the ITIL processes is the Configuration Management Database (CMDB). It is the cornerstone to the integration and sharing of information between processes. The CMDB contains information about the IT infrastructure by representing software, hardware, documents, people information (among many other IT assets) as Configuration Items (CI). In addition, the CMDB also contains information about the relationships between CIs.

Although the ITIL books provide insight and guidance on how to define CIs and their relationships, detailed methods for modeling IT infrastructure are left up to the implementers. This paper sets out to define an object model by first identifying the basic building blocks of the IT infrastructure. Then, using an example, a more detailed model is constructed step-by-step based on the principles established in the conceptual object model.

2.0 Configuration Management

Inventory and asset management provide an accounting of the entities which comprise the enterprise infrastructure, and track each entity through its own life cycle.   Configuration management encompasses both inventory and asset management. In addition, it includes establishing and maintaining the relationships between entities to provide a holistic view of the IT infrastructure to support all other IT Service Management processes.

Knowing how individual entities relate to each other, we can then determine if the failure or degradation of one entity will affect others in the infrastructure. This knowledge becomes the basis for impact analysis.

3.0 Object Model's Goals

The intent to derive an IT infrastructure model stems from the pursuit of better management of IT services. Therefore, the model should aim to accommodate the following service related goals:

- Better management and accounting of IT assets.

- Better alignment of IT and the business so that they work in partnership to achieve the enterprise objectives.

- Impact analysis of the IT environment for troubleshooting and planning.

4.0 High Level Requirements

- The model should represent not only IT assets as CIs, but also account for people and events affecting the IT infrastructure (which are captured by Service Records).

- The model should enable the representation of physical as well as logical entities. Logical entities can be used to group physical entities with similar properties or represent virtual entities (i.e. services offered to end-users, virtual servers in a server cluster environment).

- The model should describe the relationship and position of CIs for a given IT service and their relationships with other IT services if those linkages exist.

- The relationships between CIs should incorporate ITIL's IT Service Management framework and principles so that Service Management objectives can be achieved.

- The model structure should enable the integration of IT Service Management processes.

- The relationships between CIs should provide dependency information.

- Mapping of IT services to IT components should simplify IT service offering from the business point of view.

- The model structure should be flexible to accommodate definition of CIs and their relationships at a granularity suitable to an organization's level of control, the resources available and the effort required to support it.

- The model should enable enforcement of policies and contractual agreements over the usage of physical CIs.

- The model should enable dynamic changes in a given relationship between two CIs based on pre-defined conditions.

5.0 Case Scenario

The following scenario illustrates the application of the object model in resolving an incident that could occur in a real IT setting. Section 9 uses the concepts and building blocks described in this paper, to construct a detailed representation of this scenario.

1. An end-user calls the Service Desk notifying of an incident with System "A1".

2. The Service Desk Analyst opens an incident ticket and uses the CMDB to search the CI for System "A1". He associates the incident ticket with the System "A1" CI. After reviewing the incident and problem tickets associated with this System "A1" CI, he determines that this is a new incident and no workaround has been established. Based on the people CIs associated with System "A1" CI, the Service Desk analyst will then know who supports the system and assigns the incident tickets to the appropriate 2nd line support group.

3. Upon receipt of the ticket, the 2nd line support person quickly notes all the CIs that are already associated with the System "A1". He finds the technical support document that is stored in the System "A1" Document CI and uses it to troubleshooting the incident.

4. Based on the information gathered, the 2nd line support person determines that one of the processes running on the server of System "A1" needs to be recycled. In order to perform this task he needs to bring down the server.

5. Before initiating any changes, the 2nd line support person further analyzes the maintenance window of System "A1" by examining the availability window stored in the attributes of System "A1". In addition, he performs impact analysis on how this change will affect other CIs that are dependent on the server by closely examining the relationships this server CI has with other CIs.

6. His investigation reveals that another CI, System "B2", has software running on the same server. He checks the availability window of System "B2" stored within the System "B2" CI and determines that although System "A1" can be shutdown end of business day at 5:00pm, System "B2" requires the server to be running until 10:00pm.

7. He files a Request for Change ticket and associates it to the server CI and submits the RFC for approval.

6.0 Object Modeling

To understand the IT infrastructure as a collection of related entities, we begin by constructing an object model of the enterprise. Beginning with a traditional view of the IT infrastructure as a group of technologies, the model should at least encompass all the IT assets which comprise the enterprise IT infrastructure. These assets, from a company perspective, include the processes (documents), roles (human resources) and services. A service is a collection of IT resources that is provided to a customer to enable them to operate their business on a day-to-day and long term basis. The stakeholders of the enterprise or the ultimate users of the assets are then represented separately. As shown in Figure 1, this model of the enterprise (as broken down into assets and stakeholders) initially resembles the demand side of a service transaction.
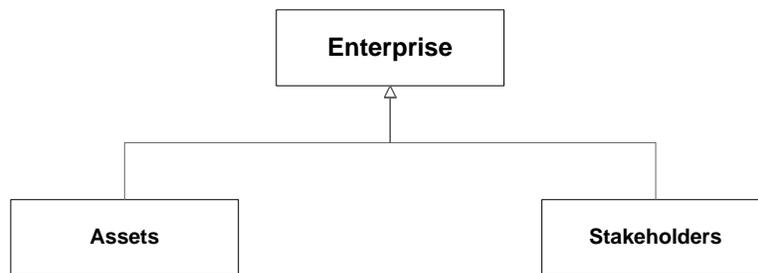
Figure 1: Model A

As stated earlier, our intent is to progress from an Asset Management approach toward a service-centric representation of the infrastructure. To align the model more explicitly with service representation, we can re-group the main entities so that the model reflects the enterprise as a transaction between two roles – a provider and a consumer of the service being provided, and the service itself. In Figure 2, the model also explicitly identifies the provider and consumer as *types of* roles, whereas roles and services are *subsets of* an enterprise. This distinction will be further elaborated in Section 8.
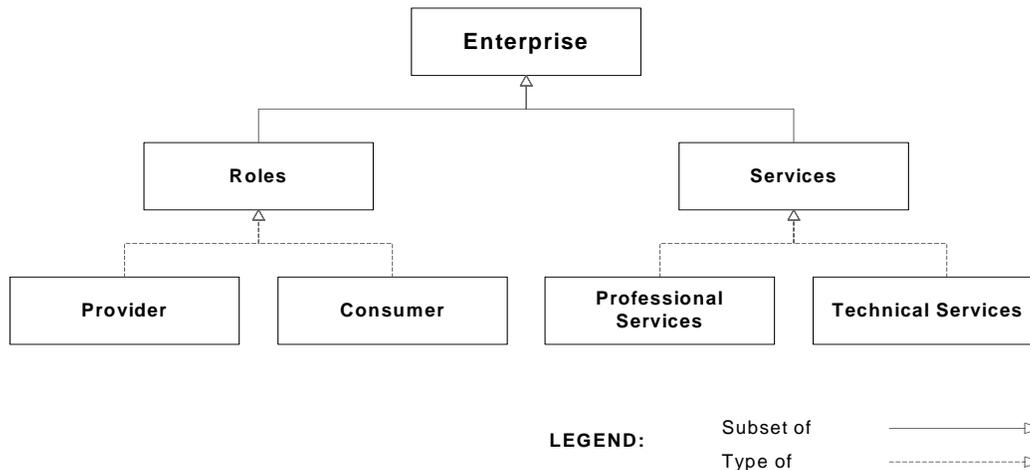
LEGEND: Subset of
           Type of

Figure 2: Model B

Now the model can be used to represent an exchange of services between generic provider and consumer companies. For IT business enterprises, the services (technical assets) can reside with the provider or consumer company. Particularly in larger organizations, some services are provided internally (company owns technical assets), while others are outsourced (provider owns technical assets). The model in Figure 3 illustrates a service that is provided by the company's own IT department and subscribed to by internal lines of business.
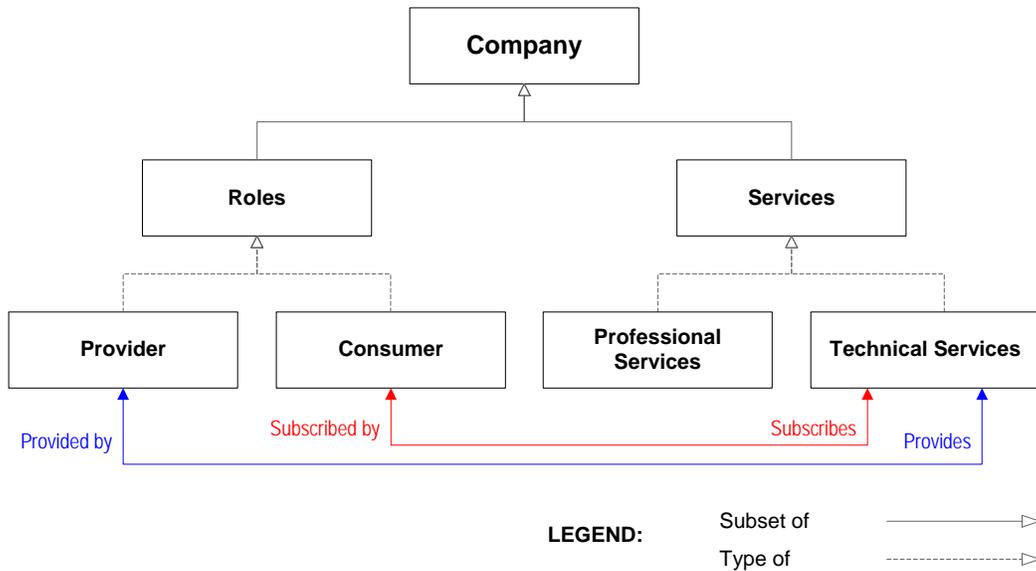

Figure 3: Model C

7.0 Configuration Items (CIs)

At this stage, before breaking down the current entities into more detail, we should establish a common terminology to apply to the model. Adopting the ITIL framework for service management, every distinct entity that is part of the infrastructure can be denoted as a Configuration Item (CI). Referring to our model, a CI can be a tangible, discrete item (such as hardware, software and documentation) which makes up a service.

From a service management perspective, anything that may be involved in the failure or degradation of a service – including the failure itself – can equally be modeled. Ideally, all the possible entities that may be represented can be classified as follows:

1. Tangible, discrete items (e.g. hardware, software, technical documentation)
2. Policies, standards, contracts that bind two or more entities
3. Roles played by persons or groups (e.g. users, service providers, support groups)
4. Events that occur at specific times (e.g. service failures/ interruptions, scheduled maintenance)

By re-grouping these classifications under roles and services, we arrive at three elemental groups of CIs: the first being technical assets – consisting of tangible, discrete items and contracts. Secondly, the roles played by persons or groups comprise another group called people. Finally, since the distinction of service interruptions (incidents) and derived service events (changes, problems, known errors) is fundamental in the ITIL framework, these events make up a third group of CIs. In our model, these events are represented by the service records that are generated to track their processing. As shown in Figure 4, the model is now composed of CIs (people, service records, technical assets) which are related to each other.
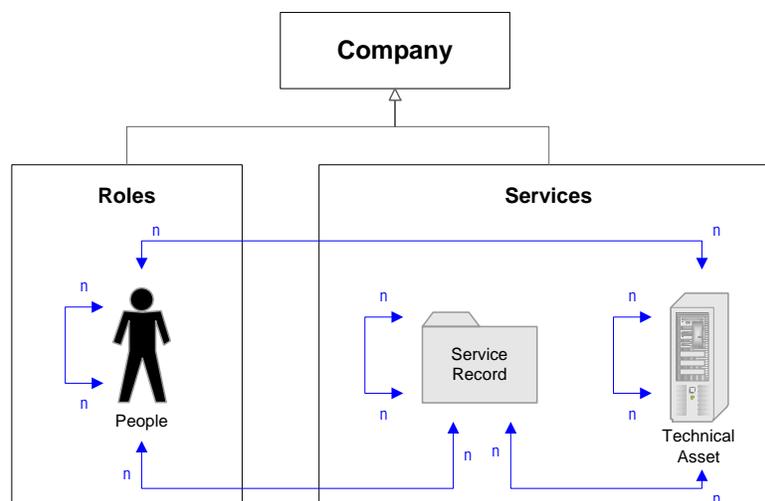

Figure 4: Model D

7.1 Logical, Physical and Virtual CIs

Note that up to this point, all the CIs discussed have been logical groupings of physical entities. These logical CIs represent grouping of CIs with similar properties, which exhibit common behavior, and have common relationships to other CIs. For example, software is a logical CI, whereas a specific business application (with version, release and patch numbers, etc.) is a physical CI. For entities which provide the same functions as physical CIs but do not have a physical form, we define them as virtual CIs. Examples of virtual CIs are virtual machines in a server cluster environment.

7.2 CI Attributes

The characteristics or properties used to describe a CI are called its attributes. Every PC can be described by its serial number, asset tag, CPU or memory size. Every person in a functional role has a name, title, login ID or contact telephone number. Every service record has a number or details to describe an event. An attribute is the abstraction of a property; if the attribute is the name then the property itself may be 'John', 'Ellen' or 'Thomas'. The distinction between a CI and an attribute may depend on the level of detail with which an organization will track its entities. A monitor may be considered an attribute of a PC if the two are tracked as an ensemble, or it may be a separate CI that is related to a PC.

8.0 CI Relationships

As depicted in Figure 4, in addition to their attributes, CIs also have relationships with each other. These relationships provide valuable information about how CIs are grouped (hierarchical relationships) and how they influence each other (functional relationships). By tracing the relationships from one CI to the next, we can gain an understanding of how an action taken on a given CI (i.e. addition, removal or modification) may influence the rest of the infrastructure and the organization.

8.1 Hierarchical Relationships

Hierarchical relationships can be used to describe the decomposition of an *aggregate* into its subsets – as we have done in the breakdown of company into roles and services. They can also be used to describe variations or types of one entity; service providers and consumers are *types* of roles. Continuing this grouping with services, we can also identify two *types* – technical services (which rely primarily on technology for their provision) and professional services (which rely more on human resources).

Once we have generated an exhaustive list of CIs, we can then group them by hierarchical relationships. A final iteration of the model (as presented in Figure 5) breaks down the roles and services into their subset CIs. For roles, (starting at the top) an organization is an aggregate of functional work groups, which are aggregates of people. For services, (starting at the bottom) various technical domains are subsets of a system, which is a subset of a service. The top tiers of this model represent logical CIs. The bottom tier represents physical and virtual CIs (both are instances of logical CIs).

Borrowing from composite industry standards, a given service can be made up of the following broad technical domains:

1. Platform
2. Software
3. Database
4. Network
5. Documentation

To complete this hierarchical breakdown, we also want to include the devices which are attached to the physical component groups and tracked as separate assets, but which cannot function on their own (e.g. peripherals, external storage devices, etc.) Thus, we will add the following entities to the model:

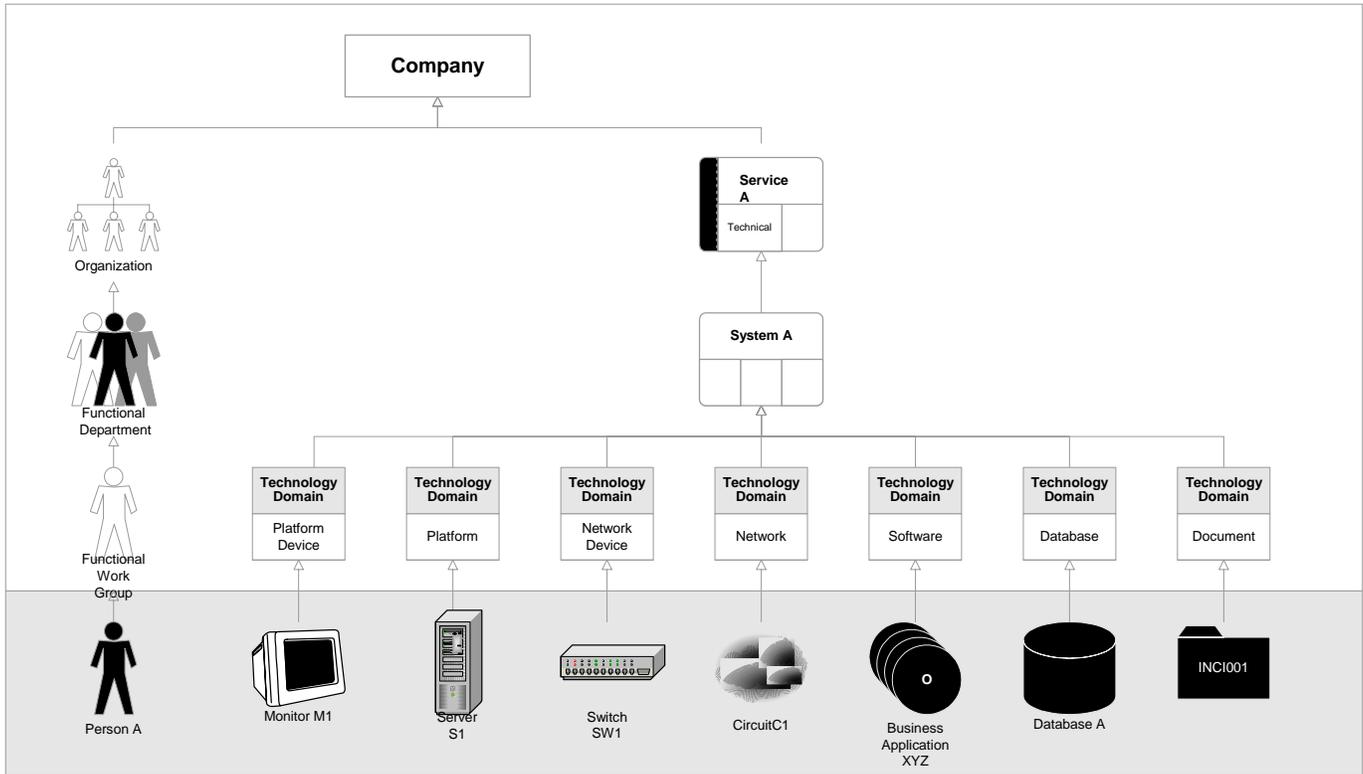6. Platform Device
7. Network Device

Figure 5: Model E

Since roles can also be broken down by aggregation, we will expand the people branches similarly to technical assets. Service providers (internal and external) and consumers are said to be aggregates of departments, which are aggregates of work groups.

8.2 Functional Relationships

Knowing only the composition of a system's individual technical domains (e.g. its platform and software entities) is often insufficient. It is equally important, for example, to know the hosting relationship of the platform to the software. In most environments, the types of hosting relationships that exist between instances (physical CIs) will be more meaningful than the relationship itself (i.e. between classes of entities or logical CIs). Therefore, to supplement hierarchical relationships (which are more meaningful at the logical level), we require another layer of functional relationships, which will represent the peer-to-peer relationships in the IT infrastructure at the physical level.

From a service perspective, we also require the model to provide information on how a service and its subset entities are ultimately impacted by service events. Having already included events in the model (represented by service records under documentation), we can relate an event to not only a service, but to any of its component entities as well – depending on what is affected (e.g. entire service, multiple systems, specific asset).

Returning to the premise of a service transaction between a provider and consumer, service records complete the picture by relating the service to people. Altogether, these functional relationships (across technical domains and across CI groupings – technical assets, people, and service records) can be summarized as follows:

1. Ownership   (e.g. owner of, submitted by)
2. Direction   (e.g. governs, uses, runs on)
3. Communication   (e.g. creates datafeed to)
4. Conditional   (e.g. A activates B if C occurs)
5. Connectivity   (e.g. connected to, installed on) – can be mapped using discovery tools

8.3 CI Relationship Types

The richness of the information extractable from an object model depends on the number of different types of relationships that is deemed meaningful:

- For simplicity, we may choose to use 'connected to' and 'used by' as the only two relationships to map all CIs in the IT infrastructure. However, this enables us to track only those CIs which have these relationships (while omitting others which do not), thus limiting our ability to perform more detailed analysis.

- On the other hand, if too many relationship types are defined, the model may become too complex to use. It may also be cumbersome to keep all the relationships current for every CI modification.

There should be a balance between the need for more information and ease of use. The following tables outline some key relationships between various types of CIs (hierarchical and functional), as identified in Figure 4:

1. Technology assets to technology assets   (Table 1)
2. Technology assets to people   (Table 2)
3. Technology assets to service records   (Table 3)
4. Service records to service records   (Table 4)

The relationship types presented are not meant to be exhaustive. Depending on the perceived business value and the effort required to maintain and update the relationships, one may choose to shorten or expand the list.

Before proceeding with the list, we should point out one more convention in notation. In depicting CI relationships, a reciprocating relationship is implied. For example, the figure below can be worded in one of two ways:

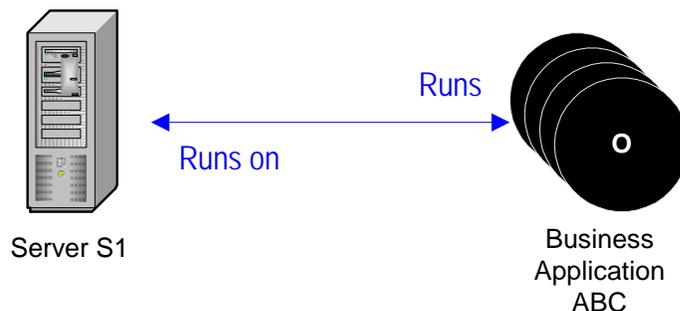- "Server S1 **Runs** Business Application ABC" or "Business Application ABC **Runs on** Server S1"

Server S1 — Runs / Runs on — Business Application ABC

Table 1: Technology Asset to Technology Asset

Relationship	Relationship Type	Usage
Assembly Used when one CI is a component of another or a CI is a subset of another CI	Subset of / Aggregate of Member of / Consists of Component of / Assembly of	• Linkage between a physical CI to a logical CI • Linkage between two logical CIs • Linkage between a physical CI and a virtual CI. • Linkage between a physical CI which is a component of another physical CI.
Physical/Network Connectivity Describes the network and device to device connectivity between two hardware CIs	Connects to / Connected to	• Linkage between two physical CIs which are connected via a physical medium such as a wire cable or fiber optics.
Hosting Application Data Dependencies Identifies where a software or database CI is hosted or installed on another CI. Shows the exchange of data between two systems or within a system	Runs / Runs on Creates datafeed / Receives datafeed Runs (contains) / Runs on (installed on) Uses / Used by Contains / Installed on	• Linkage between a software CI and a platform CI which shows where the software is being executed only. The storage of the executable is located on another platform CI. • Linkage between a software CI and a database CI which shows where the software is being executed and stored as well. • Linkage between a software CI and a platform CI which shows where the software is stored only. The execution of the software is on another platform CI.

Table 1: Technology Asset to Technology Asset (Continued)

Relationship	Relationship Type	Usage
	Contains / Installed on	• Linkage between a software CI and a platform CI which shows where the software is stored only. The execution of the software is on another platform CI.
Governance Describes the enforcement of policies, contracts, regulations, process or procedures over the operations or used of another physical CI	Limits usage of / Usage limited by	• Linkage between a document CI and the physical CI over which it has legal or regulatory control.
	Supports / Supported by	• Linkage between a document CI and the physical CI for which it provides process or procedural instructions.
Conditional Describes the dynamic relationships between two CI based on predefined conditions	Activates / Activated by	• Linkage between any two physical CIs - which is active only under specific conditions

Table 2: Technology Asset to People

Relationship	Relationship Type	Usage
People Describes the roles between people and the Technology Asset	Subscribes to / Subscribed by Provides / Provided By	• Linkage between a System CI and the person who uses it or the person who provides it
	Supports / Supported by	• Linkage between System CI and the person who supports it.

Table 3: Technology Asset to Service Record

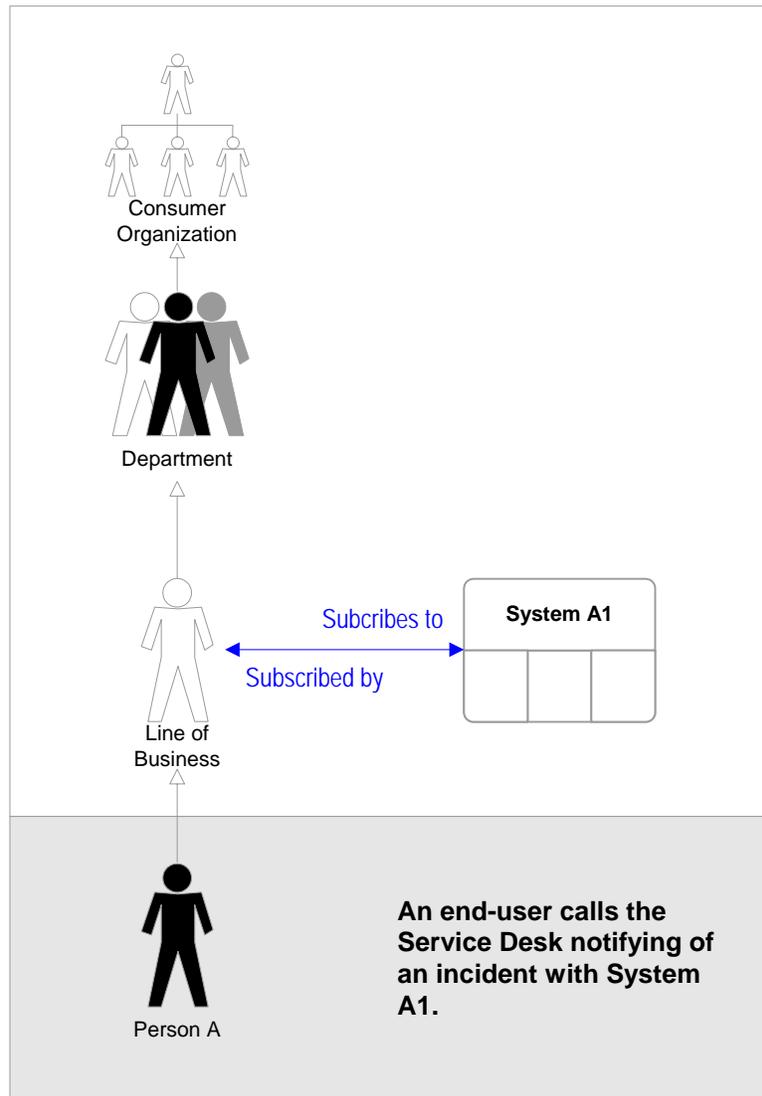| Service Record	Relationship Type	Usage
Incident Record	Impacts / Impacted by	• Linkage between CI and service record to capture service outage or user issue.
User Request Record	Performs / Performed on	• Linkage between CI and service record to capture user's request to modify/add/delete (MAD) a CI. Usually low risk change.
Problem Record	Examines / Examined by	• Linkage between CI and service record to capture of the root cause(s) of an incident
Known Error Record	Causes / Caused by	• Linkage between CI and service record to capture permanent fix information.
Change/Release Record	Modifies / Modified by Fixes / Fixed by	• Linkage between CI and service record to capture change or release instructions.

Table 4: Service Record to Service Record

Service Record	Relationship Type	Service Record
Incident Record	Duplicate of / Duplicated by Relates to / Related By Causes / Caused By	**Incident Record**
	Examined by / Examines Requires workaround / Provides workaround	**Problem Record**
	Requires workaround / Provides workaround	**Known Error Record**
	Resolved By / Resolves Causes / Caused By	**Change Record**
Problem Record	Duplicate of / Duplicated by	**Problem Record**
	Initiates / Initiated by Identifies / Identified By	**Known Error**
	Provides Workaround / Uses Workaround	**Change Record**
Known Error Record	Provides Fix / Uses Fix	**Change Record**

8.4 CI Relationship Attributes

To enhance the understanding of a relationship between two CIs, attributes about the relationship can also be defined. For example, suppose that the relationship between a Service CI and the IT Support group is "System X is supported by Support Group A". By assigning an attribute to the relationship, we can capture whether the support is provided by "1st Line" or "2nd Line" support. The use of a relationship attribute in this case reduces the number of relationships required to define various levels of support available to System X.

9.0 Application of the Model

Returning to the case scenario presented earlier in Section 5, we will now apply the model, via the relationships between the CIs involved.

Service C

Professional

System A1

Incident Mgmt.

Impacts

Provider Organization

Production Support

Technology Domain

Document

Service Desk

Impacted by

INCI003

Submitted by

Submits

Person B

The Service Desk analyst opens an incident ticket and uses the CMDB to search the CI for System A1. He associates the incident ticket with System A1.

System A1

Impacted by

INCI001 — Examined by → PBM001

Examines

Impacts

INCI002

Examines

After reviewing the incident and problem tickets associated with this System A1 CI, he determines that this is a new incident and no workaround has been established.

System A1

Supported by

Supports

Assigned to

UNIX Support

INCI003

Assigned group

Person C

Based on the people CIs associated with System A1, the Service Desk analyst then knows who supports the system and assigns the incident ticket to the appropriate 2nd line support group.

Service A

Technical

System A1

Technology Domain	Technology Domain	Technology Domain	Technology Domain	Technology Domain
Document	Software	Database	Platform	Network Device

Technical Document D1

Business Application ABC

Database A

PC A

Server S1

Switch SW1

Runs (contains)

Runs (contains)

Runs on (installed on)

Connected to

Upon receipt of the ticket, the 2nd line support person quickly notes all the CIs that are already associated with the System A1. He finds the technical support document that is stored in the System A1 document and uses it to troubleshoot the incident.

Service A

Technical

Service B

Technical

System A1

System B2

Technology Domain

Software

Technology Domain

Platform

Technology Domain

Platform

Technology Domain

Software

O

Business Application ABC

Server S1

Server S2

O

Business Application XYZ

Receives datafeed

Creates datafeed

His investigation reveals that another CI, System B2, has software running on the same server. He checks the availability window of System B2 stored within the System B2 and determines that although System A1 can be shut down end of business day at 5:00pm, System B2 requires the server to be running until 10:00pm.

He files a Request For Change ticket and associates it to the server CI and submits the RFC for approval.
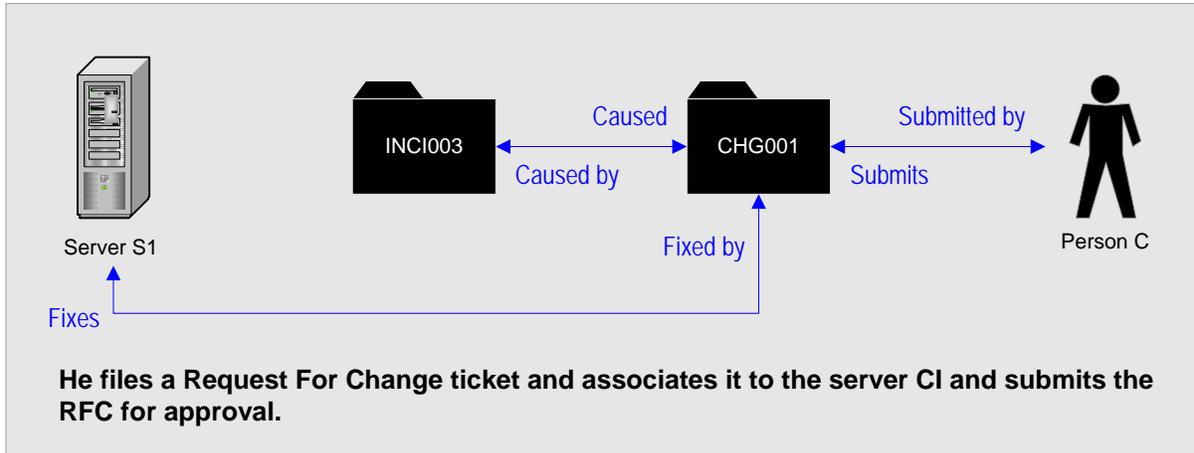
10.0 Conclusion

Although the example provided depicts a relatively simple application of the model, the same principles can be applied to more complex environments. Even this example, when all the relationships are laid out, begins to hint at the complexity of mapping entire infrastructures – as illustrated in Appendix A.

Appendix A