Flyer

opentext™

Application Modeling-as-a-Service

Create a solid foundation for IT management in the service-oriented world.

Key Benefits:

- **Better Service Delivery:**
  As discussed, many such capabilities rely on and can benefit from application modeling.
- **Focus on Your Core Mission:**
  Your IT staff are not modeling experts and you don't necessarily want to maintain a permanent skills base to allow you to continuously discover and model applications.
- **Flexible Consumption Model:**
  You pay per application for as many applications as you decide to model.

Overview

It's All About Services

Users are shifting their consumption models. They are no longer interested in consuming components, and they increasingly expect a true service-oriented experience. Simply put, they want a service but they are not interested in what is required to deliver it. The concept of a service is becoming more central to how IT must think and operate, and IT must shift from managing by technology domain to managing by service supporting the business.

But what constitutes a business service? While there is no simple answer, one thing is clear—business services are supported by IT services, mainly applications. Therefore, managing applications is a necessary stepping stone to managing services. Applications have a real manifestation in both the collection of hardware and software elements that make them up and the dependency relationships between these components. Being able to model applications—that is, map all elements of the application and their relationships and interdependencies—is vital to IT's ability to make this shift.

Building the Model

Creating the model is a combination of data collection and mapping. Data collection is usually done using discovery, but is sometimes also augmented with data import. Each discovered component (such as server or software element) becomes a configuration item (CI) and is stored in OpenText Universal CMDB® (UCMDB). Discovery can identify many relationships between the discovered components (such as database X is contained in server Y) and these too get stored in UCMDB to form the core of the model. But not all relationships are discoverable. Mapping refines the model by adding relationships that cannot be discovered (such as the association of an application to a service). The model can then be visualized, typically in a tree-like format. Once you have defined the model, it too is stored in UCMDB. Based on the model, you can then drive all of the lifecycle management functions, from design to operation—including costing, change and configuration management, and more.

Key Features

The Model: the "Swiss Army Knife" of IT Management

The fact that you cannot manage something unless you are able to measure it is a known management principle. This can be applied to the management of an application: Unless you know what it is made up of, it is unlikely that you will be able to effectively manage it. Having a model is crucial to the context of almost any IT service management use case. Here are some of the most common examples:

Monitoring and Service Level Management

The same way a doctor needs to know the anatomy of the human body to monitor its health, you need to know the anatomy of an application to monitor its health. When things go wrong, it is the model that allows you to remediate the situation with minimal business impact by allowing you to propagate status and determine application health by aggregating the health of its underlying components. So when you are experiencing issues, it is easy to determine and resolve the root cause. Last but not least, the service model provides the ability to both define and measure underpinning contracts (UCs), operating level agreements (OLAs), and service level agreements (SLAs).

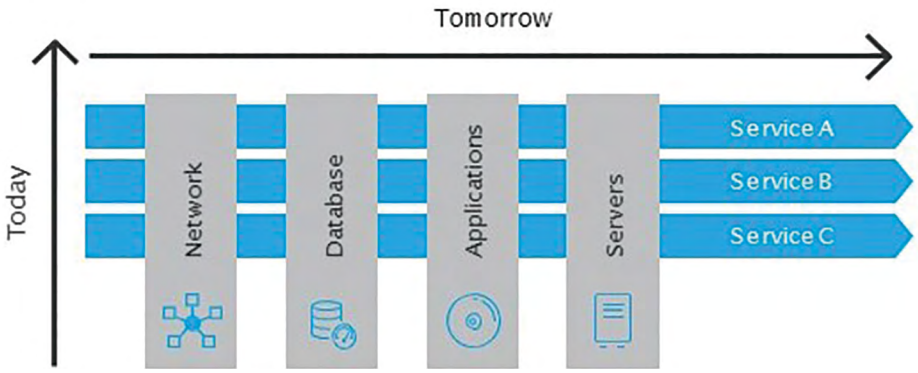

Figure 1. The shift from managing technology stacks to managing services

Change Management

A key component of any change process is impact analysis. Before you make changes to the production environment, you want to be able to determine what those changes may impact and how. A model facilitates and streamlines impact analysis by showing you how application components are related so that you can run "what-if" simulations to determine and mitigate any adverse impacts from the proposed changes.

While most IT organizations have configuration and compliance standards for components such as databases or operating systems, it is rare for such standards to exist for applications or services. Managers typically have no visibility into the configuration state of their applications, nor can they tell with confidence whether they are compliant. Considering the scale and complexity involved, it is not feasible to inspect each component and then somehow aggregate this data into an application-wide picture. A model can easily identify the make-up of an application and the state of each component, enabling a holistic view of the total state and quick identification and remediation of any non-compliant elements.

Cloud and Application Financial Management

One of the expectations that business users have, as a result of cloud adoption, is a move to a consumption-based financial model, with full transparency of the total cost of ownership per application. Because they can get this from public cloud providers, users are now asking IT to provide the same even for non-cloud applications. An application model allows you to enumerate all of the assets that make up an application, without which it is not possible to provide this level of visibility.

Service Planning and Deployment

OpenText Application Modeling-as-a-Service from OpenText Professional Services is a consumption-based offering: rather than pay for person-time, you pay for an outcome. At the conclusion of the service, we deliver to you the number of application models you have purchased.

What Is Application Modeling?

Application modeling is the process of describing in visual form the relationships between various IT infrastructure components (such as servers, J2EE applications, and databases) and how together a collection of such components or CIs comprise an application. The application can then also be mapped to a business service that it plays a part in delivering.

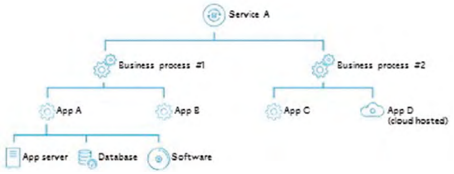

Figure 2. Sample model

Applications are tiered according to size: bronze, silver, gold and platinum. The size determines the price for each map. The main objective is to define for each application a map that dynamically updates itself as components are added or removed from the application. To do this, we follow this process:

Application Onboarding

In this step, we populate your UCMDB with server CIs. There are generally two basic methods to do so: discovery and import. You may already have discovered some servers but to ensure that no gaps exist, we ask you to list all servers belonging to the application and we import this list into the UCMDB. We then validate that your discovery can, from this point forward, reliably continue to discover not just all of the servers and their attributes but also the software that is running on them and its attributes. Each element we discover constitutes a CI.

Static Mapping

Once we have collected all of the CIs that make up an application in the UCMDB, we use them to create the application map. If you wish to also map to business services, we then associate the application with one (or more) business

Connect with Us

services (if needed). We also ensure that your discovery is configured to continuously collect and update the data relevant to the application and all of its components. This completes the mapping process and reflects the application configuration at this point in time.

Dynamic Mapping

Applications rarely remain static, and unless steps are taken to update the map, it will quickly become outdated. To prevent this, we use a technique called "fingerprinting." It ensures that the map remains current by dynamically adding and removing components based on the results from continuously discovering your servers.

Key Benefits

Better Service Delivery

As discussed above, many such capabilities rely on and can benefit from application modeling. Being able model your applications accurately and consistently will allow you to mature and improve your capabilities, ultimately leading to better service delivery.

Focus on Your Core Mission

Your IT staff are not modeling experts and you don't necessarily want to maintain a permanent skills base to allow you to continuously discover and model applications. Engaging OpenText Professional Services relieves you from this need and lets you leverage our expertise so that you can focus on using application models to execute your primary tasks, be they monitoring, change management, or any other activity.

Flexible Consumption Model

You pay per application for as many applications as you decide to model. We classify applications into tiers—bronze, silver, gold, and platinum—depending on scale and complexity. This allows you to allocate a clear cost for each application, and provides you with the security of knowing that you pay only for what you need.

The OpenText Professional Services Difference

OpenText provides unmatched capabilities with a comprehensive set of consulting and implementation services and unique intellectual property that help you drive innovation through streamlined and efficient software delivery:

- Proven OpenText software solution implementation expertise
- More than 20 years of experience helping large,complex, global organizations realize value from their OpenText software investments
- Rich intellectual property and unparalleled reach into product engineering
- Technology-agnostic implementation approach with no vendor lock-in, no rip-and-replace
- Education and support services to ensure adoption

Learn more at
www.microfocus.com/en-us/professional-services/overview

www.opentext.com

261-000237-001 | O | 11/23 |

opentext™