Title: Implementing a Spatial CMMS Solution using Maximo 7

Presentation Abstract: City of Austin's Watershed Protection Department contains the Drainage Utility. It is responsible for maintaining a network of aging stormwater infrastructure, altered channels and ponds which capture, redirect and treat stormwater runoff. While Austin has a reputation for being a big city which retains a small town feel, it's infrastructure issues are definitely those of a big city. Unfortunately, its CMMS capabilities were geared towards a small town, composed primarily of disparate Access databases, institutional knowledge and outdated maps to facilitate day to day operations. To modernize and move from a reactive to a proactive stance, a COTS asset management system (Maximo) was implemented to replace the legacy databases, capture and share institutional knowledge and provide field access to GIS. A key requirement for this new system was spatial integration within the CMMS to support crews utilizing laptops. This paper discusses project requirements, current system capabilities, configuration/workflows, and project challenges.

Biography: Philip Campman has worked in the Austin Water/Drainage Utilities for over twenty years. Bachelor of Arts in Geography from the University of Texas at Austin.
Biography: Andrew Rudin has worked Watershed Protection Department for the last 5 years. Previously he worked as a GIS consultant in civil engineering and for the State governments of Texas and Florida. Bachelor of Science in Geography and International Affairs from Florida State University.
Biography: Jeremy Wolfe of GeoNexus Technologies has worked on Maximo and Esri GIS integration for 10 years. Bachelor of Science in Computer Science from University of Wisconsin – River Falls.

Overview

The City of Austin Texas has been one of the fastest growing cities in the country for many years due to its high quality of life, relatively low cost of living and a diverse and dynamic economy that embraces a high tech job market while retaining its small town charm. According to the 2010 U.S. Census, Austin as a city had a population of 790,390 residents with 1,783,519 people within the Austin-Round Rock-San Marcos metropolitan area. It is the fourth largest city in the state of Texas.

Naturally, a large city such as this must provide many services to its citizens. The City of Austin is one of the top five employers in the city with over 10,000 employees. Basic services such as libraries, health clinics, police, fire and EMS are provided to its citizens. In addition, it must create and maintain a significant amount of physical infrastructure to facilitate provision of other basic services. The City owns and operates a combined water/wastewater/reclaimed water utility known as Austin Water Utility (AWU), an electric utility known as Austin Energy (AE), Austin-Bergstrom International Airport (ABIA) and a stormwater utility (Watershed Protection Department) as well as Public Works Department (PWD), Austin Transportation Department (ATD) and Parks and Recreation Department (PARD).

Maintaining vast amounts of physical infrastructure is obviously very difficult without tools for tracking work. AWU has used Infor's Hansen v7.x for maintenance tracking for over a decade. AE uses a combination of products including IBM's Maximo for its work tracking. ABIA implemented Maximo as well. All of these systems however are stand-alone; they are not shared across the City for various reasons. For example, ABIA and Austin Energy must maintain their own instances of Maximo due to Homeland Security regulations. Other departments have relied upon relatively crude methods for maintaining work requests including Access databases and Excel spreadsheets. While work has been getting done, it has been done usually from a reactive rather than a proactive stance. Also, no coordination of activities among these departments has been possible without significant manual analysis.

The Watershed Protection Department (WPD) began a project in 2006 to implement a modern Enterprise Asset Management (EAM) system to replace its antiquated collection of Access databases and spreadsheets. In parallel, WPD also began work on a multi-million dollar project in 2006 to create a Geographic Information System (GIS) of its entire physical infrastructure including pipes, inlets, manholes, manmade channels, headwalls and ponds comprising the manmade stormwater conveyance network. The goal was to unite the EAM system with the GIS to provide a modern spatially-based system.

Since the Maximo product was already in the process of being deployed at Austin Energy and ABIA, WPD did a gap analysis of its business requirements to the Maximo product to determine if it was a good fit. This was done with the assumption that WPD would implement the new Spatial component of Maximo to provide the integration between Maximo and the GIS. The Maximo Spatial product appeared to be a good fit for WPD's business needs. WPD agreed to implement Maximo Spatial as a Citywide enterprise application which could eventually be shared with other departments as they became ready. In late 2010, EMA Inc. was brought on board as an external consultant to help the City implement Maximo Spatial v7.1.x. The project team consisted of technical personnel from both WPD and the City's Communications and Technology Management (CTM) department, non-technical subject matter experts (SME) from WPD, and EMA consultants and sub-consultants.

In addition to the business process diagrams, functional requirements and other documentation about WPD's business, the City held a workshop with potential future users of a Citywide Maximo system to create set of Enterprise standards. (See Appendix A: Citywide Standards for Asset Status.) These standards were critical to ensure that the system as configured would not be uniquely configured to support just the first user (i.e. WPD) but would be usable by the other departments that followed. The standards also drove some data changes in the GIS to make it more compatible for integration with Maximo.

The team began working on the project in November 2010 and WPD Field Operations went 'live' with the system in August 2011. Since then a second department, ATD, has begun using the system as well. The City is in the process of adding another WPD work

group to the system (Pollution Prevention and Spills Reduction) while in parallel upgrading to v7.5 Maximo Spatial. A third department, PWD, recently completed an RFP process to select a vendor to implement their processes in the Enterprise Maximo as well. Waiting in the wings are other departments including Emergency Medical Services, Fire Department and Building Services. WPD, ATD and PWD will take full advantage of the Maximo Spatial capabilities by integrating it with existing GIS databases.

Server Environment Configuration and Software Versions

One major lesson learned during this project was to develop a clear picture of the hardware configuration needed to support an Enterprise Maximo system. At the beginning of the project it was clear that the hardware needed to support the base Maximo installation had been procured, upon closer examination of the desired Enterprise environment, significant gaps in hardware were discovered. Once examined, additional hardware resources were specified and purchased during the course of the project. (See Appendix B: Maximo Production Infrastructure Diagram)

Maximo Spatial 7.1.1 was introduced with Maximo 7.1.1.6 in December 2009. This version of Maximo Spatial was rewritten to simplify the map interface and leverage the new ArcGIS Server REST API. These changes made it much easier to implement and support Maximo Spatial with regard to both complexity and hardware resources. The City was able to take advantage of this new version of Maximo Spatial, since they began their implementation on Maximo 7.1.1.6. This version of Maximo Spatial requires an ArcSDE 9.3.1 database and ArcGIS Server 9.3.1. Since the City was planning on upgrading their enterprise GIS environment to ArcGIS 10.0, a publishing process was established using Safe Software's Feature Manipulation Engine (FME) to "publish" features from the ArcSDE 10.0 databases to this ArcSDE 9.3 database on a nightly basis.

The Maximo and ArcSDE databases use Oracle 11g Release 2 on separate Linux servers. The Maximo application servers are running in IBM WebSphere 6.1 on Windows Server 2003 R2. ArcGIS Server 9.3.1 for .NET is also running on Windows Server 2003 R2.

Best practice for an Enterprise software configuration requires at least three parallel server environments to support Development, Test and Production, respectively. Each server environment should be equivalent in terms of server allocation i.e. each server performing a function in the Production environment should have an equivalent within the Test and Development environments.

Obviously there needs to be a Production environment which the users access day to day to perform their work. A Development environment separate from Production is necessary so that future configurations and system changes can be developed over the course of time without affecting the daily use of the Production environment. It should match the target (Production) environment in terms of architecture (although system performance may or may not match).

A Test environment should mirror the Production environment to reflect both architecture and performance. As new configurations and system changes are completed in

Development, they can then be migrated to the Test environment for user acceptance testing and regression testing to ensure that existing functionality hasn't been adversely affected. Equivalency of resources between Production and Test ensures that the performance results seen on Test adequately reflect what is to be expected in Production.

Maximo Spatial as configured by the City is a very complicated architecture. It is composed of several different components, each of which must be represented in Production, Test and Development. In addition, the City chose to operate Maximo in a Websphere Application Cluster as well as Oracle Data Guard to provide significant fault tolerance. Therefore, not only is there architectural mirroring through a Production/Test/Development environment, there are redundancies associated with providing a fault tolerant environment. The Production environment is replicated in two different locations within the City to create a primary and failover site.
The main components of the Production environment include:

1. One HTTP Web Server
2. One Maximo application cluster containing four application servers across two virtual machines for vertical and horizontal clustering of Java Virtual Machines (JVMs)
3. One Maximo application server for a standalone Maximo integration/cron task application
4. One Oracle database server for the Maximo database
5. One Oracle database server for the ArcSDE database and ArcSDE instance
6. One ArcGIS Server
7. One Safe Software FME Publishing Server

Some of these servers are configured as stand-alone servers configured as a Windows Virtual Machine (VM) (1, 2 and 3). Others are configured as virtual machines within an AIX LPAR (4 and 5). The remainders reside on physical machines dedicated solely to perform functions supporting the Maximo system (6 and 7). To provide the fault tolerance, all of these production servers (virtual and physical) are duplicated at a second remote site and connected using a combination of Websphere clustering, Oracle Data Guard and BIG-IP load balancing technology, as appropriate.

Additionally, there are GIS data entry servers used within the City that communicate with the FME server in order to publish new GIS data to the Maximo GIS instance.

Asset, Location, Feature Class and Service Address

Integration of Maximo and GIS is highly desirable when attempting to use Maximo Enterprise Asset Management of utility assets. Utilities (e.g. WPD which is a drainage utility) typically have thousands, if not hundreds of thousands, of assets such as pipes, manholes, inlets and other physical infrastructure. Base Maximo is a tabular system. That is to say implementing base Maximo 7.1.x results in a web application that can be used to track work orders, service requests, inventory and so forth however it lacks any spatial (GIS) functionality. Maintenance on assets such as manholes or pipes could certainly be tracked in such a system however populating assets within Maximo to which work may be tracked would be difficult each asset would need to be entered by hand. Even if a complete inventory of assets were to be imported to base Maximo, it would still

be difficult to sort through the thousands of records to find the exact pipe or inlet to which a work order should be attached for example.

The main point of Maximo Spatial is to extend base Maximo through the provision of spatial capabilities. Spatial is an add-on to base Maximo which provides three types of key functionality:

1. Allows Maximo to utilize an ESRI ArcGIS SDE database for the provision of Maximo Assets, Locations, Service Addresses, Service Requests and Work Orders which correspond to each feature contained within an SDE feature class or classes.
2. Provides an integrated map display within the Maximo web application to display Maximo assets spatially and provide an interface for user interaction using map tools for selection, query, digitization, etc.
3. Leverages existing GIS databases containing addressing information by exposing this information within Maximo as native service addresses and for geolocation capabilities within the map display.

As discussed above, the Maximo server architecture contains both a Maximo database and a GIS database. The key to integration of these two databases using Spatial is to expose the GIS feature classes as tabular views at the database level and then consuming these views within Maximo as native tables. When properly consumed, the table views can represent a collection of Assets within Maximo or a list of valid addresses. Maintenance of the GIS features can be done outside of Maximo and through the use of a Maximo Cron task, resynced so that Maximo and the GIS remain consistent.

Providing geographic references through Spatial opens many windows of opportunity for users. For example, when a spatial location is associated with a work order, the user can zoom to the location on the Maximo Map and see the asset in context with its surroundings and other assets or work being performed nearby. There are four different ways of attaching a geographic reference to service requests and work orders using Spatial:

1. Asset Feature Classes
2. Location Feature Classes
3. (Native) Feature Classes
4. Service Address Feature Classes

From a utility management perspective, Assets within Maximo typically represent discrete pieces of infrastructure that are treated as a single unit such as a curb inlet or a pipe segment - ‘hard’ infrastructure against which you want to track maintenance costs or time spent repairing/using. WPD treats manholes, inlets, pipe segments and such as separate assets since each is repaired and/or replaced as a whole. When defining an Asset spatially, a GIS feature class in the Maximo SDE instance is exposed to Maximo and then defined as an Asset Feature Class. When creating a work order against an Asset, you first choose the type of Asset Feature Class (inlet, manhole, etc.) and then use the Maximo Map to pick the individual Asset using spatial selection tools.

Locations on the other hand typically represent a place which contains one or more physical pieces of infrastructure. Its primary purpose is to group together Assets (typically to roll up the costs for all Assets associated with the Location) or to simply associate seemingly disparate Assets located at the same parent Location. An example would be a drainage pond Location. Since WPD has defined ponds as individual Assets and then associated them with parent Locations, a parent/child relationship has been established between the pond assets that exist (sedimentation, filtration, etc.) at a single related Location. While they may be different pond Assets, the whole Location can be seen as a unit to roll up costs for maintaining all of the Assets at that Location.

Feature Classes can be defined to exist within Maximo natively rather than be maintained externally. This can be useful to allow a user to edit the geography of a feature using the Maximo Map tools rather than editing the feature outside of Maximo. A common example of this would be to define a polygon feature class called WOPOLY in which users could digitize polygon features to represent the extent of work to be performed in a work order. These native Feature Classes are not associated with an Asset or Location but rather represent more an area of interest for a work order or service request.

Service Addresses are point features exposed to Maximo similar to an Asset or Location however they are unique in that they explicitly represent a discrete address point rather than a piece of infrastructure or a general place. As such, they contain both a spatial component (a point in space) as well as a tabular component (address text). Service Addresses are also used within the Maximo Map for geolocation by address.

When assigning geography to a Maximo service request or work order, it is quite possible to mix and match any or all of these types of geographic references together on the same record. For example, a field worker may create a work order to do maintenance on only a section of a large filtration pond:

- The pond Asset that will be worked on is chosen using the Maximo Map and associated to the work order.
- Multiple ponds may exist at that site and be related to each other, therefore a pond Location is associated with the work order to allow the lookup of the related ponds at that site.
- A Service Address is specified on the work order representing the street address to which the field crew should be dispatched to access the pond.
- Finally, a polygon is digitized into the WOPOLY Feature Class representing the precise section of the Pond that needs work.

The Maximo Map utilizes a function called 'autolocate' which centers the Maximo Map on the spatial location associated with the work order when the user switches to the Map tab. Since multiple spatial representations can be associated with a service request or work order, Maximo employs a hierarchy to determine which spatial representation to autolocate when switching to the Map tab:

a. Native Feature class (e.g. WOPOLY)
b. Location
c. Asset
d. Service Address

It is highly desirable to have a spatial location representing all work performed. If no spatial location has been associated with a work order, there is no ability to show it on the Map. Since Maximo has four options for this and each option could be the most correct one, it is unfeasible to make any of them mandatory. Instead, users are trained to associate a spatial location with every work order they produce. In addition, site administrators occasionally run queries to look for work orders that do not have a valid spatial location assigned and send them to supervisors for correction.

GIS Data Preparation on Editing System

GIS Data Created and Maintained Natively in Maximo

Maximo Spatial allows users to create and manipulate GIS records from within Maximo. The Spatial Edit tool supports the creation and manipulation of GIS features associated to Assets, Locations, Service Addresses, service requests or work orders. Through signature security, users are primarily limited to creating and manipulating service request and work order features, since external maintenance SDE GIS instances are the system of record for all service addresses and most assets and locations.

Four feature classes were created in the Maximo SDE instance to contain spatial representations of service requests and work orders: SRPOINT (service request points), SRPOLY (service request polygons), WOPOINT (work order points) and WOPOLY (work order polygons). To support both Maximo users and GIS users that don't have access to Maximo, it was determined that each feature class containing features generated by and maintained from Maximo would contain copies of the associated data in the Maximo records. For example the description, status, work type and various other data are copied from a work order to its associated feature automatically by Maximo when a work order is changed. Crossover domains are used to copy the data initially when the feature is created by Maximo. Then the Maximo Integration Framework (MIF) is used to propagate the data to GIS whenever a work order associated to a GIS feature is updated. Finally, Maximo escalations are in place to update GIS data that wasn't properly copied from Maximo to GIS. This is done for redundancy, since various activities might prevent a change from propagating through the MIF. Examples are that a change occurred when the system was in Admin Mode, the publish channel capturing the change was disabled for testing purposes or the external system receiving the change was disabled for testing purposes.

Service request feature classes are configured in the same way as work order feature classes. These configurations exist for work order point and polygon feature classes as well as service request point and polygon feature classes. Erosion sites are Maximo locations that are entirely maintained in Maximo, including the GIS feature. The erosion site feature class is configured in much the same way as service request and work order feature classes, where edits to the location in Maximo propagate to GIS. This approach of copying data from Maximo records into the GIS feature class is preferred to the use of ArcSDE views, because performance remains consistent over time as records are added to Maximo. The performance of the equivalent ArcSDE views across a database link degrades over time as records are added to the system.

GIS Data Created and Maintained Externally but Synchronized with Maximo Spatial

Most GIS feature classes that need to be shared with Maximo are too complicated to be edited and stored natively within Maximo as a native Feature Class. These feature classes typically represent the bulk of a utility's assets. Traditional maintenance of these feature classes is performed by trained GIS technicians who not only create the spatial representation for features but also invariably add multiple attributes. Maximo may even be a secondary use for these feature classes; they may be primarily used for map making, hydraulic model input and other non-Maximo uses.

WPD has several different feature classes which it creates and maintains external to Maximo. It has chosen to integrate some of these with Maximo as GIS Objects to be used as Assets or Locations. These include:

- PondAssets – Drainage ponds (polygon)
- PondLocations – Drainage pond parent locations (polygon)
- FbridgeInlet – Bridge inlets (point)
- FcomboInlet – Combination inlets (point)
- FcurbInlet – Curb inlets (point)
- FDischargePump – Pond discharge pumps (point)
- FGrateInlet – Grate inlets (point)
- FHeader - Headwalls (point)
- FManhole - Manholes (point)
- Inlet_Cleaning_Areas – Defined areas for inlet cleaning routes

Most of these feature classes have multiple attributes associated with them. Most of them also have a related table which contains hyperlinks to reference documents such as field photos, videos and other documents. (These related tables are referred to generically as REFDOCs.)

These feature classes are stored and edited on the WPD maintenance SDE instance. Specialized data entry tools exist to add or change features in these feature classes. Any changes to them are routed through the WPD GIS staff including any changes that are made in the field. (A process for addressing these field changes is outlined below.)

Two data changes needed to be made to each feature class to prepare it for Maximo. First, hyperlink values in the REFDOC tables needed to be recalculated to make them compatible with Maximo. Previously these hyperlink values were in the format:

*server**share**filename.ext*

These hyperlinks needed to be recalculated to the following format:

file:////server/share/filename.ext

Figure 1shows an example of a hyperlinked photo that is associated with a curb inlet asset. Thousands of these photos are available within Maximo as hyperlinks.

Figure 1

Second, values in the STATUS field were made consistent with the STATUS values that were agreed upon for Maximo when establishing Citywide standards. (See Appendix A.)

In addition, a change to the editing process needed to be made. Normally, when an asset is abandoned and removed in the field, it would be deleted from the GIS as well. With the integration of Maximo and GIS, it was no longer desirable to remove these features from the GIS since there may be historic work records associated with them. Since WPD instituted the policy of having a spatial representation for every work order in Maximo,

deleting the asset in GIS would effectively remove the spatial representation for any work orders associated with the removed asset. Therefore, the GIS entry process was altered to require setting the STATUS to "Decommissioned and Removed" for any features that would have previously been deleted.

GIS Data Created Externally Once and Then Maintained Natively in Maximo Thereafter

It is not always necessary to edit GIS data outside of Maximo and pull it in. Maximo spatial provides the ability to create new GIS records for assets, locations, and work orders directly through a web map interface.

Some GIS data used in Maximo does not have as complicated a data entry process as the stormwater utility or the ponds data and in such cases, the overhead of maintaining the data on a maintenance server and synchronizing the edits to Maximo would be excessive. The one day time lag between GIS edits made by users to when the new records would be available in Maximo can also be factor.

One of the WPD groups having data loaded into Maximo is the Stream Restoration Section. This work group tracks erosion issues along creeks within the City, designs plans for remediation of these problems, and then manages construction projects based on these plans. Before Maximo implementation, their data was stored in a Microsoft Access database, and consisted of three main tables and two shapefiles:

1. Erosion Complaints - A table that stores calls from citizens concerning the location of erosion problems. Reports from City staff who find erosion problems during field work are also tracked here. It contains citizen information and details of the problem. The spatial location of these records is a street address.
2. Erosion Sites – A table storing information about physical locations along creeks where problems have been reported. Once a call is received about a problem, the Stream Restoration Group visits the site and assesses the condition, and creates a record in this table. It contains the address of the problem, and details about the problem, details about the creek geomorphology, and the severity of the problem. One erosion site could have many complaint records tied to it. Erosion site records were associated with erosion site points in a shapefile.
3. Erosion Projects- A table storing information about projects to fix the issues identified in the Erosion Sites table. Contains information about how the problem will be fixed, rough timelines for construction, and contact info for project managers. One erosion project can fix several erosion sites in the table above. Erosion project records were associated with erosion site polygons in a shapefile.

In addition, the erosion sites table utilized complex code to score the severity of sites based on different physical properties of the site, such as the width of the creek, the slope of the banks, the soils in the creek bed, etc. When values for these properties have been entered or changed, the editor could use a tool in the Access database that analyzes these values and runs calculations that populate several condition assessment fields on the table. These values are then used to prioritize construction projects.

The shapefiles contained many attributes redundantly tracked in the Access database, and the data in both was rarely kept in sync. Some records existed in the Access database tables that did not exist in the shapefiles, and vice versa.

In order to prepare the GIS data for loading into Maximo, missing records were identified in both tables, and the Stream Restoration Group either added or deleted records on either side until there was a one to one relationship between the GIS records and the Access records. Since the Access database attributes were considered more up to date than those in the shapefile, all GIS attributes were removed, leaving only the unique ID that joins the GIS records to the Access records. The Access tables were joined to the two GIS shapefiles in ArcMap and exported to file geodatabase feature classes with the appropriate attributes. These two feature classes were then loaded into the Maximo SDE. These feature classes and Maximo then became the system of record for both tabular and spatial representation of Erosion Sites and Erosion Projects following Maximo go live.

Erosion Projects were loaded into Maximo as work orders associated with a spatial location in the newly created Erosion Projects Maximo feature class.

Erosion Complaints were loaded into Maximo as service requests. The spatial component for each was derived from the address stored in the table. They were associated with Service Addresses in Maximo. In addition, Erosion Complaints were associated with a parent Erosion Site Location. When you switch to the Map tab on the Erosion Complaint service request, it centers the map on the Erosion Site Location rather than the Service Address associated with the complaint.

Service Addresses

Service Addresses are critical to the call center process of creating service requests. It is not the responsibility of the call taker to know which Asset or Location should be attached to a service request. Instead the call taker asks the caller for an address or intersection near the issue they are reporting, and the call taker chooses the nearest Service Address from the map. This allows the call taker to complete the call as quickly as possible and an investigator to determine the actual infrastructure that needs to be repaired at a later time.

Address points depicting all physical building addresses within the City are maintained in a GIS separate from Maximo. A combination of escalations and integration framework components are used to keep the Service Addresses within Maximo in sync with the address points in GIS. This includes loading data into multiple Maximo sites and updating Maximo Service Addresses when an address changes in GIS.

Pros and Cons of Storing GIS Externally vs. within Maximo

All GIS data used in Maximo is stored within the Maximo SDE database regardless of whether it is created and maintained external to Maximo or within Maximo utilizing the Maximo Map tools. There are pros and cons as to why you would pick one maintenance methodology over the other.

While it is possible to integrate a maintenance SDE instance with Maximo and edit these feature classes directly using ArcMap, it is not recommended for a few reasons. First, performance issues may result of such an arrangement. Editing the feature classes could slow down the performance of the SDE instance and consequently affect the performance of Maximo. Conversely, since Maximo Spatial references the Maximo SDE feature classes for different purposes (including display on the Map tab), peformance may be adversely affected for the GIS editors. In addition, these GIS feature classes may be used by others outside of Maximo for map production, Internet viewers, analysis, and other uses which may also cause performance issues.

Second, feature classes in a maintenance environment tend to be more dynamic than those used within Maximo. Adding new fields, changing domain values, adding or dropping feature classes and other activities are fairly common on an active maintenance SDE. Maximo GIS SDE in an Enterprise environment needs to be much more tightly governed. GIS administrators have more flexibility if the maintenance GIS SDE and the Maximo GIS SDE are separate instances.

If the choice is to use separate instances for Maximo GIS SDE and maintenance SDE, a synchronization method must be employed to keep the two instances synchronized. (See Data Updates Utilizing FME below.)

Table 1 lists the pros and cons when deciding to store a feature class within Maximo as a native Feature Class or to store a feature class on a maintenance SDE instance and synchronize the data to Maximo. (It assumes that the Maximo GIS SDE and maintenance GIS SDE are separate instances.)

	External Feature Class	Native Feature Class
Supports editable related tables	Yes	No
Geometry can be edited within Maximo	No	Yes
Attributes can be edited with Maximo	No	Yes
Custom editing tools can be used	Yes	No
Synchronization tool needed to keep Maximo GIS current	Yes	No
Supports versioned editing scenarios	Yes	Yes
Must utilize cron task	Yes	No

Table 1

Storing feature classes externally provides much more flexibility to GIS administrators and editors. However, for simple feature classes for which maintenance by Maximo users is desirable, a native Feature Class is a good option.

Maximo SDE preparation

Copy SDE data from Editing SDE to Maximo SDE(Andrew)

The dedicated Maximo SDE provides a single location for GIS data utilized by the application as Assets, Locations, or Service Addresses. These same GIS layers are used by the ArcGIS map services that drive the Maximo embedded maps. When loading GIS data from the maintenance SDE to the Maximo SDE it is important that GIS feature classes share the same schema on each instance. Having one schema simplifies maintenance of both databases and makes syncing of data between the servers less time intensive and avoids the need for special tools to translate fields.

Add required fields

After each feature class was copied to the Maximo SDE, four new Maximo specific fields were added and populated accordingly:

1. MXASSETNUM for Asset layers, MXLOCATION for Location layers or MXADDRESSCODE for Service Address layers (String). These values are generated from an Oracle sequence that was seeded at an extremely high numeric value, so it wouldn't generate the same values as the sequences used by Maximo to auto-number ASSETNUM, LOCATION and ADDRESSCODE values.
2. MXSITEID (string). Populated with value 'WPD' for all assets. This filters these assets so they are only visible to Watershed Protection Department users.
3. MXCREATIONSTATE (integer). Populated with the value 1 for all records pending loading into Maximo.
4. ROWSTAMP (integer). Populated from an Oracle sequence whenever a record is edited. This is used by Maximo to detect concurrent edits.

Create triggers

A trigger was added to each of these feature classes to set the values of these fields. Whenever a record is inserted in one of the feature classes, the following field values are set:

1. MXASSETNUM, MXLOCATION or MXADDRESSCODE: Initialize the value from the shared Oracle assetnum sequence. This will be the ASSETNUM, LOCATION or ADDRESSCODE value loaded into Maximo when the associated record is created.
2. MXSITEID: set to a value of WPD. This will be the SITEID value loaded into Maximo when the associated record is created.
3. MXCREATIONSTATE: Set to a value of 1, which indicates that Maximo should create the associated record.
4. ROWSTAMP: Intialize the value from the shared Oracle rowstamp sequence. This will allow Maximo to update the record. If this value is NULL, Maximo cannot edit the record.

Create table views for Maximo

Since the feature classes are stored within a separate Oracle database than Maximo (i.e. the Maximo SDE instance), they are accessed across a database link. It is required by Maximo Spatial that these feature classes have a SHAPE column that utilizes the ST_GEOMETRY geometry storage type. This is an Oracle data type that utilizes LOB storage. LOB columns cannot be accessed across a database link using SELECT statements, so views must be created for each of these feature classes to effectively hide the SHAPE column from the Maximo application.

Maximo Spatial uses the dbms_sql package to update the SHAPE column for these feature classes when the user creates or edits a feature's geometry from within Maximo. The ArcGIS REST API is used to read the geometry information from the associated map layer.

Maximo GIS data loading

With the exception of the Erosion site feature class, assets and locations are created for each new feature in GIS by the PlusSFeatureLinkCronTask cron task. This cron task is part of Maximo Spatial and provides a means to create assets or locations from GIS features. Originally, one cron task instance was configured for all location GIS objects and another was configured for all asset GIS objects. This proved to be problematic when troubleshooting why individual features were not loading into Maximo. Now there is one cron task instance for each asset or location GIS object to allow us to troubleshoot the load for a single feature class.

The mxe.pluss.cron.maxrecords.per.execution system property is set at 4,000. This limits each execution of a cron task instance to a maximum of 4,000 loaded records. This is important, since some of the feature classes contain over 100,000 records. Attempting to process 100,000 records during a single execution of the cron task instance presents 2 problems.

1. Each record being processed must be loaded into memory in the Maximo JVM as an individual Maximo Business Object (mbo). This alone could consume a lot of memory.
2. The cron task instance executes very rapidly, consuming a considerable amount of CPU time. Limiting the number of records ensures that the cron task executes in a short amount of time, freeing CPU resources for other processes.

Each of the cron task instances was set to execute every 10 minutes during the initial load of data. This ensured that all GIS data was loaded within a couple of days. After the initial load of each feature class, its associated cron task was reconfigured to execute every hour. The reason it is run multiple times per day is to allow features to be loaded into Maximo throughout the day to publish data from the maintenance SDE geodatabases to the Maximo SDE geodatabase.

GIS Object Configuration

To register a GIS feature class in Maximo, a GIS object must be defined in Database Configuration. A GIS object is defined as such by the following attributes:

1. The Java class name is psdi.pluss.mbo.SDEMboSet
2. The unique column is OBJECTID
3. Two key attributes are defined in the object

Maximo Entity	Key Column 1	Key Column 2
ASSET	MXASSETNUM – UPPER (12)	MXSITEID – UPPER (8)
LOCATIONS	MXLOCATION – UPPER (12)	MXSITEID – UPPER (8)
PLUSSSERVICEADDRESS	MXADDRESSCODE – UPPER (12)	MXORGID – UPPER (8)
SR	MXTICKETID – UPPER (20)	MXCLASS – UPPER (16)
WORKORDER	MXWONUM – UPPER (10)	MXSITEID – UPPER (8)

4. The optional MXCREATIONSTATE attribute is defined for asset and location feature classes from which associated records are created in Maximo by the PlusSCreateFeatureLinkCronTask cron task.
5. The Add Rowstamp checkbox is selected to allow Maximo to edit the records in GIS.

Any additional attributes that are exposed through Maximo search dialogs or conditional UI sections on Maximo screens are also configured for each GIS object.

After the table is created in the Maximo database by applying configuration changes in Database Configuration, the table is renamed by prefixing it with GIST_, so it can be restored later if necessary. Then a synonym is created with the same name as the GIS object that references the feature class across the database link. This effectively tricks Maximo into accessing the remote GIS feature class when it queries or updates records in the GIS object.

Next, a relationship must be created on the Maximo object that is to be associated with the GIS object. For example, if the GIS object is named FBRIDGEINLET and is to be associated with assets, a SPATIAL_FBRIDGEINLET relationship is created on the ASSET object where the child object is FBRIDGEINLET. The following table shows example relationships for each of the spatially enabled Maximo objects. The naming convention of the relationship is critical. It must begin with the prefix SPATIAL_, followed by the name of the child object as shown in Table 2.

Parent Object	Child Object	Child Object	Where Clause
ASSET	FBRIDGEINLET	SPATIAL_FBRIDGEINLET	mxassetnum=:assetnum and mxsiteid=:siteid
LOCATIONS	EROSION_SITE	SPATIAL_EROSION_SITE	mxlocation=:location and mxsiteid=:siteid
PLUSSSERVICEADDRESS	ADDRESSES	SPATIAL_ADDRESSES	mxaddresscode=:addresscode and mxorgid=:orgid
SR	SRPOINT	SPATIAL_SRPOINT	mxticketid=:ticketid and mxclass=:class
WORKORDER	WOPOLY	SPATIAL_WOPOLY	mxwonum=:wonum and mxsiteid=:siteid

Table 2

Finally, the GIS object must be registered in the GIS Admin (Spatial) application. This enables association of the object to one or more map layers in the Map Manager (Spatial) application. It also creates signature security options to allow users to select or change the Feature Class value of an asset, location, etc. to the GIS object value. For example, granting the WOPOINT signature option in the work order Tracking application to a security group allows users to select WOPOINT as the Feature Class in which to create a feature to associate to a work order. The signature option is only granted to the MAXADMIN security group when it is automatically created.

Maximo Map Administration

A key component to Maximo Spatial is the Map tab that is accessible in key contexts within Maximo. (The Map tab should be configured only after GIS Object configuration has been completed.) This map is driven by services provided through ArcGIS Server. The provisioning of these services is, in general, fairly straightforward. An MXD is created representing the content that is desired for the Maximo Map. The MXD is published as a service on an ArcGIS Server instance. Once it is published, the map service is configured and consumed within Maximo and presented on the Maximo Map tab. There are considerations that need to be taken into account when configuring the Map.

At least one base map service should be configured that utilizes a tile cache. This not only improves drawing performance, but it greatly aids users in navigating the map. Users can use the zoom levels provided by the tile scales to allow zooming in and out of the map while maintaining a consistent center point. When a user is viewing a record linked to GIS within one of the Maximo applications that is spatially enabled, clicking the Map tab will highlight the associated feature and center the map on that feature. Using the zoom scales allows the user to zoom in and out while keeping that feature at the center of the map. Figure 1 shows the Maximo Map tab which has autolocated a curb inlet Asset highlighted by the large yellow star.

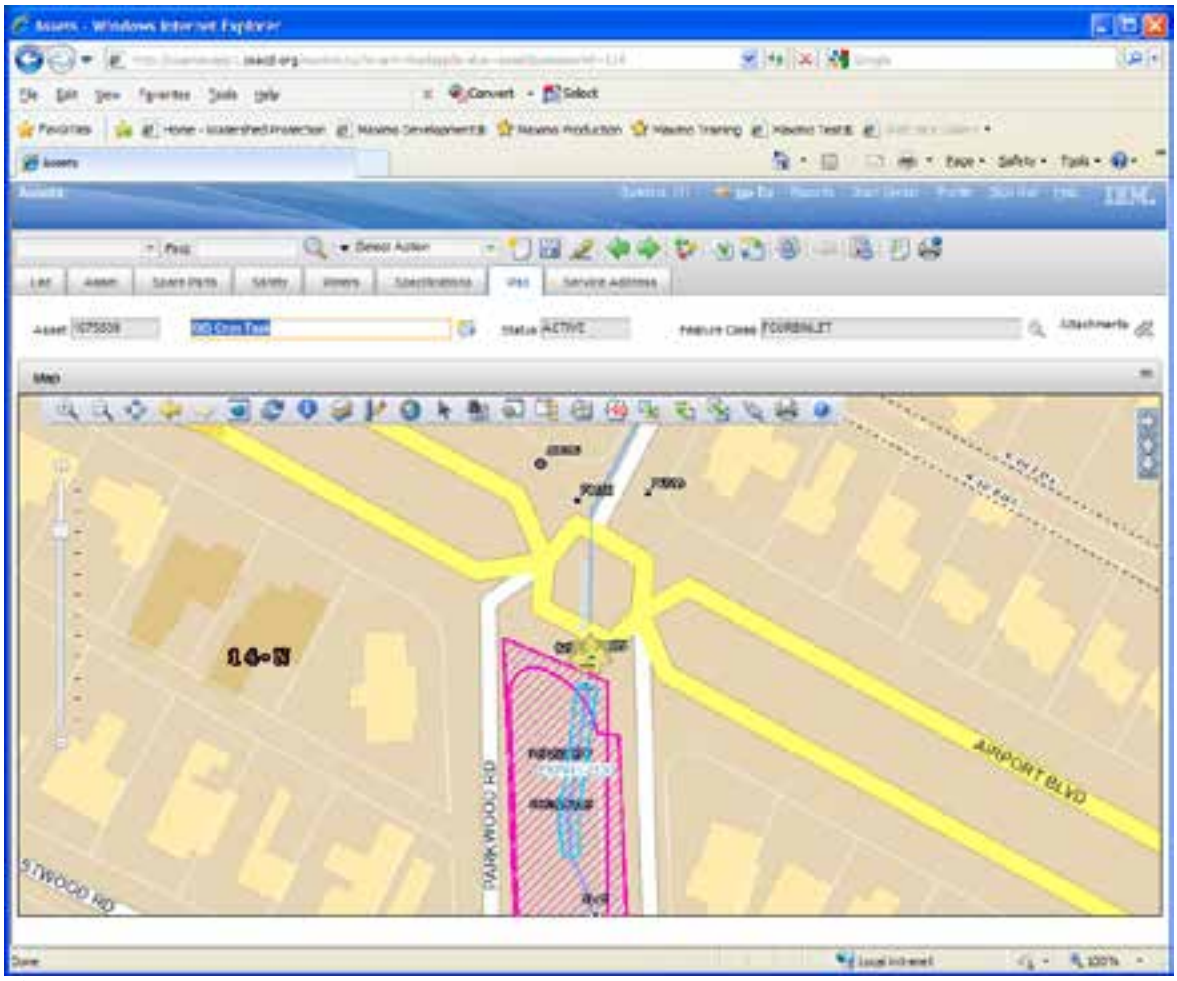


Figure 2

One Site Versus Many Sites

Core functionality within Maximo is the ability to designate Sites. A Site is a logical grouping within Maximo into which users, processes, access rights and other configuration details are separated. Users choose a default Site in which to work and that defines what types of work orders, Assets, Tools, etc. that that user can manipulate. For example, a work order created in Site A cannot be assigned to a member of Site B. (Service requests act independent of Site and can be shared across Sites.)

Deciding whether to utilize a single Site or multiple Sites within Maximo is one of the key decisions that must be made when establishing an Enterprise instance. If it is critical that work orders be shared across all work groups, then implementing multiple Sites would cause difficulties. The City chose to implement multiple sites since each Site at the City would represent a different department. The separation of department work orders is actually an intended benefit of choosing that route.

Deciding on a single Site or many Sites also has a direct impact of the Maximo Map presentation. In Maximo 7.1.x, configuration of the Map is done at the Site level. Therefore, if you choose to go the route of a single site, you will be limited to a single

Map. If you choose to employ many Sites, each Site will have its own unique Map. Whichever type of functionality is chosen, it is still possible to configure the map to suit the needs of users, it just means different configuration strategies.

If using a single Site, the MXD document that will be used to drive the Map will need to contain every layer that is needed by every work group using the Enterprise system. Since not every layer is of interest to every work group, security within Maximo must be employed to restrict the display of layers to only the work groups that are interested. While possible to do, it would mean a very complicated maintenance strategy. And since the configuration of the security rights would be at the system level, it would need to be done by a centralized Maximo support person rather than department-level admins. There are also performance considerations involved when trying to scale an MXD of this size to serve all of the work groups. Scale issues could be avoided by breaking the MXD down into several MXDs and layering the services however the issue of security management still remains.

With a many Site approach, each Site has its own Map. Configuration of the Map is done at the Site level and since little security is involved, it can be delegated to local department Site admins.

MXD Preparation

As mentioned previously, ArcGIS services can be layered with the Maximo Map to produce a single Map interface. This is necessary for several reasons:

- Using multiple smaller MXDs to create ArcGIS Server services provides better performance than one very large MXD which drives a single service.
- Transparency is controlled at the service level irrespective of the transparency setting for a layer within an MXD.
- By splitting layers into separate MXDs, it is possible that a common basemap service may be employed with Site unique services layered on top.

WPD uses the following MXDs to create services. They are layered with the first service as the 'top' service as shown in Table 3:

WPD_WO_SR.MXD	All open work orders and service requests
WPD_Assets.MXD	All of the feature classes that are exposed within Maximo as WPD Assets
COA_Base_Addressing.MXD	Contains address points and street centerlines
WPD_Base_PointsLines.MXD	Additional basemap layers of interest to WPD (points and lines)
WPD_Base_Polygons.MXD	Additional basemap layers of interest to WPD (polygons)
2009_02_6IN_TC_Aerial.MXD	Raster service containing 2009 6" true color aerial photography
COAViewer_dev.MXD	Common cached raster basemap theme

Table 3

MXDs are created and maintained by the WPD site administrator. Symbology for each map layer in the MXD is what is displayed within the Maximo Map. Best practice for layering services is to segregate polygon layers into an MXD separate from points and lines since there are transparency issues to deal with. (See below.) Like any ArcGIS Server service, simple symbology leads to better performance. It is recommended that the ArcMap Analyze Map tool (located on the Map Service Publishing toolbar) be run prior to creating services to check for issues in the MXD that might negatively affect performance.

There needs to be at least one MXD that contains the feature classes which represent Assets and Locations within Maximo. This allows the user to use Map tools to select Assets or Locations on the map and return the results to Maximo.

Transparent vs. Non-transparent Services

As stated previously, ArcGIS Server does not honor transparency of layers as set in the MXD using the Adjust Transparency button on the Effects toolbar. Rather, transparency is set at the Map Service level using the Maximo Manager (Spatial) configuration screen. It is easier to place all of the base layers that need to be transparent within the same MXD since the adjustment within Maximo will affect all layers in the Map Service.

Determining the best transparency level can be simulated in the MXD that will have a transparency set by using the Adjust Transparency tool in the MXD to adjust each layer to the same level of transparency. This will approximate what it will look like in the Maximo Map when the same level of transparency is set on the Map Service. ArcGIS Server will simply ignore the individual transparency levels when the service is created.

WPD created a separate MXD (WPD_Base_Polygons.MXD) to contain reference layers that are polygon-based. In the Maximo Map Manager configuration screen, this service was applied a 66% transparency which seems to work well.

Transparency can also be controlled on a layer by layer basis within the Maximo Map. However, any transparency adjustments are only seen by the individual user and are not retained from session to session.

Map Service Definition

Once map services have been created within ArcGIS Server, the Maximo Map Manager (Spatial) configuration screen is used to compile them into a map. Then,

1. Choose Go To > Administration > Map Manager (Spatial)
2. Press the Insert Map button to create a new map
3. Fill in the Long Description for the Map
4. Type in the URL for the Geometry Service
5. Choose the Site to which the Map will be associated
6. Under Map Services, choose New Row and fill in the Name, Order, URL and % Transparency for each Map Service to be added.

The Name specified will be the text value used in the Map layer listing. The order in which the Map Services are layered is controlled by the Order field. The lower the

number in the Order, the "higher" it will be placed on the map. It is possible that additional Map Services will be inserted in the future so it is best to leave gaps in the Order numbers (i.e. 0, 5, 10, 15, 20, etc.).

In addition, for a Map Service that contains feature classes that are exposed as Assets or Locations within Maximo, each layer needs to be specified in the Service Layers list.

1. Press the Select Layers button
2. In the Select Layers dialog, select each layer that represents an Asset or Location in Maximo and press OK. These Layers will be added to the Service Layers section of the Map Service Configuration
3. Open the Detail Menu for each GIS Object entry box and choose Select Value
4. Click on the appropriate GIS Object that corresponds with the Service Layer

Once all of the Service Layers have been defined, press the Save button to save the Map configuration.

The Services tab of the Map Manager (Spatial) configuration page is used to specify the Geocode Services to be used within the Maximo Map. The Geocode Services drive the geocoding function provided using the Locator:Addresses tool in the Maximo Map. Press New Row to add a Geocode Service and then specify a Name and URL which points to the geocoding ArcGIS Server service.

Finally, the Auto Create tab of the Map Manager (Spatial) configuration page is used to define which native Feature Classes are used when the Auto Create action automatically creates and places GIS features on the map for work orders and service requests. For work order, a feature class and label need to be specified. (For example, WPD specifies WOPOINT and STATUS for the WO Feature Class and WO Label, respectively.) Also, a feature class and label need to be specified for service request. Finally, the check box to activate or inactivate this functionality needs to be checked. Press the Save button to save the Map configuration.

Security

Once Map Service Layers have been configured, security groups must be authorized to visualize the Map layers that were added to the Service Layers section during configuration. In the Security Groups appication, the group to be granted access needs to be selected. On the Application tab in the Applications windows, click on the application that uses the Map Service Layer. In the table windows at the bottom, select Grant Access next to the Layer and press Save Group.

Data Updates Utilizing FME

Once GIS Assets and Locations have been loaded into Maximo SDE from maintenance SDE, a process is needed to add new records and update existing ones. This was necessary for feature classes that are maintained on a maintenance SDE instance external to Maximo. FME is software developed by Safe Software Inc. which provides flexible GIS data interoperability tools. The FME software was purchased by the City primarily

for its ability to synchronize data between GIS layers on multiple SDE databases. The Maximo project was the first production implementation of this tool.

Assets and Locations are updated daily on WPD's maintenance SDE. FME uses a configuration file known as a "workbench" to store combinations of tools that read, modify, and write data between sources. For Maximo, each feature class associated with a Maximo Asset or Location has its own workbench. Two fields on each feature class are used in determining synchronization status between the Maximo SDE and the maintenance SDE:

1. DRAINAGE_ID: A unique numeric ID is assigned by the GIS to each feature in every feature class that is used as an Asset or Location feature class within Maximo.
2. FME_DATE: A timestamp is applied to this field whenever changes are made to a feature and at initial creation.

Each FME workbench has the same basic steps:

- Compare the WPD SDE feature class to the Maximo SDE feature class. Use the unique DRAINAGE_ID and FME_DATE fields on both layers to match records to each other.
    - New features are those WPD records whose DRAINAGE_ID does not exist in the Maximo feature class
    - Updated features are those WPD records which have a matching DRAINAGE_ID in the Maximo feature class, but the feature's FME_DATE field contains a newer date than the Maximo record.
- Once these differences are identified, FME copies these records into memory on the server for processing. At this point the new and updated features branch into two separate processes.
    - New Features
        - These records are separated in memory and given new fields to store Maximo specific values.
            - The MXASSETNUM is added and each record gets a null value
            - The MXCREATIONSTATE is a numeric field. New Assets are given a value of 1 so Maximo can identify them for processing later
            - The MXSITEID is added and each record gets a null value
            - The MXROWSTAMP is added and each record gets a null value
        - FME then appends these records to the Maximo SDE feature class.
        - Oracle triggers on the Maximo SDE feature classes then populate the MXASSETNUM, MXSITEID and MXROWSTAMP fields.
    - Updated Features
        - Once the workbench identifies records that match between both servers, this subset is run through another comparison to find records where the FME_DATE field value is greater than (newer) than the Maximo SDE record's value.

- Featuress that meet the criteria above have their MXASSETID, MXCREATIONSTATE and MXSITEID values populated from their corresponding Maximo record.
- FME then updates the Maximo feature with the maintenance SDE values for that feature.

Once the Maximo SDE has been edited by FME, new features are still not available to Maximo users. Each feature must be added to Maximo as an Asset or Location. To automate this task, EMA configured Maximo cron tasks for each feature class. The cron tasks run every 10 minutes. They search for SDE features with an MXCREATIONSTATE value of 1. Maximo creates a new Asset or Location in its ASSET or LOCATION tables, respectively, from these features. It then resets the MXCREATIONSTATE value on the features to 0.

For updated records, usually nothing additional needs to done in the Maximo system after FME runs, with the exception of status value changes. The status of an Asset or Location (whether it is proposed, active, decommissioned, etc.) is critical to how an Asset or Location can be used in Maximo. Since the GIS is the database of record for this attribute, it is maintained by GIS staff on the maintenance SDE instance. An additional cron task for each GIS layer runs every 10 minutes and searches for Maximo GIS records where the status is different from the Maximo Asset or Location's status and changes the Maximo status to match. The cron task will only update status in a forward chronological order. (For example, it won't change an active Asset back to a proposed status.)

Changing an Asset or Location's status to a decommissioned state is irreversible and places severe restrictions on its use. It is recommended that Assets and Locations are not automatically changed from Active to Decommissioned status. Instead these differences should be identified by running Maximo saved queries which determine Maximo Assets and Locations which have a status that differs from the corresponding GIS feature. These can then be manually updated after user verification.

Using Children Work Orders for GIS updates

System infrastructure is continually changing as field workers make repairs and upgrades. Inevitably, the information stored in the GIS will begin to differ unless these changes are communicated back to the GIS maintenance staff so that the GIS can be adjusted to reflect these changes. WPD has implemented a work flow within Maximo for capturing these field changes. When a work order has been completed which requires a change to the GIS, a child work order is created and assigned to the WPD GIS group.

On the child work order, a long description is added which summarizes what work was completed. The Parent work order is specified so that the GIS technician can review it if necessary. A Work Type and Work Priority are chosen as appropriate. An attachment may be provided which sketches out the changes that need to be made.

Once the GIS has been altered to reflect the new system configuration, the work order can be closed.

Future Enhancements

DataMart Publishing from Maximo SDE

GIS feature classes representing the stormwater system are maintained on an SDE instance that has limited access. The feature classes are published to a Citywide data mart for general use in mapping and analysis. When feature classes are synchronized from the maintenance GIS SDE to the Maximo SDE, the data on each server is virtually identical in terms of schema except the Maximo feature classes contain four additional fields: MXASSETNUM, MXSITEID, MXCREATIONSTATE and ROWSTAMP. Of particular interest is the MXASSETNUM which is helpful for staff trying to do maintenance analysis outside of Maximo. WPD plans to revise its publishing methodology to publish these feature classes from Maximo to the City data mart in the future so that this field will be available for general use.

In addition, there are some native Feature Classes in Maximo that are of interest to staff who do not have access to Maximo. These features will be published to the Data Mart as well.

Work Order GIS layers

Maximo is not designed to internally support complex spatial analysis of service requests or work orders. Nor does it do a very good job of displaying currently open service requests and work orders on the Maximo Map. To address both of these concerns, WPD is currently working with a consultant to implement four new feature classes within the Maximo SDE instance created through Oracle spatial views:

1. Open Service Requests
2. Open Work Orders
3. Closed Work Orders
4. Closed Service Requests

The spatial views are created by a geoprocessing script that queries the service requests and work orders based on the STATUS field. A spatial component is added in the form of a point representing the Asset, Location, Feature Class or Service Address to which the service request or work order is associated. (A point feature is used as the lowest common spatial denominator since one feature class cannot hold both point, lines and polygon features.) To determine the representative point for a line, the midpoint of the line is chosen. To determine the representative point for a polygon, the centroid of the polygon is chosen.

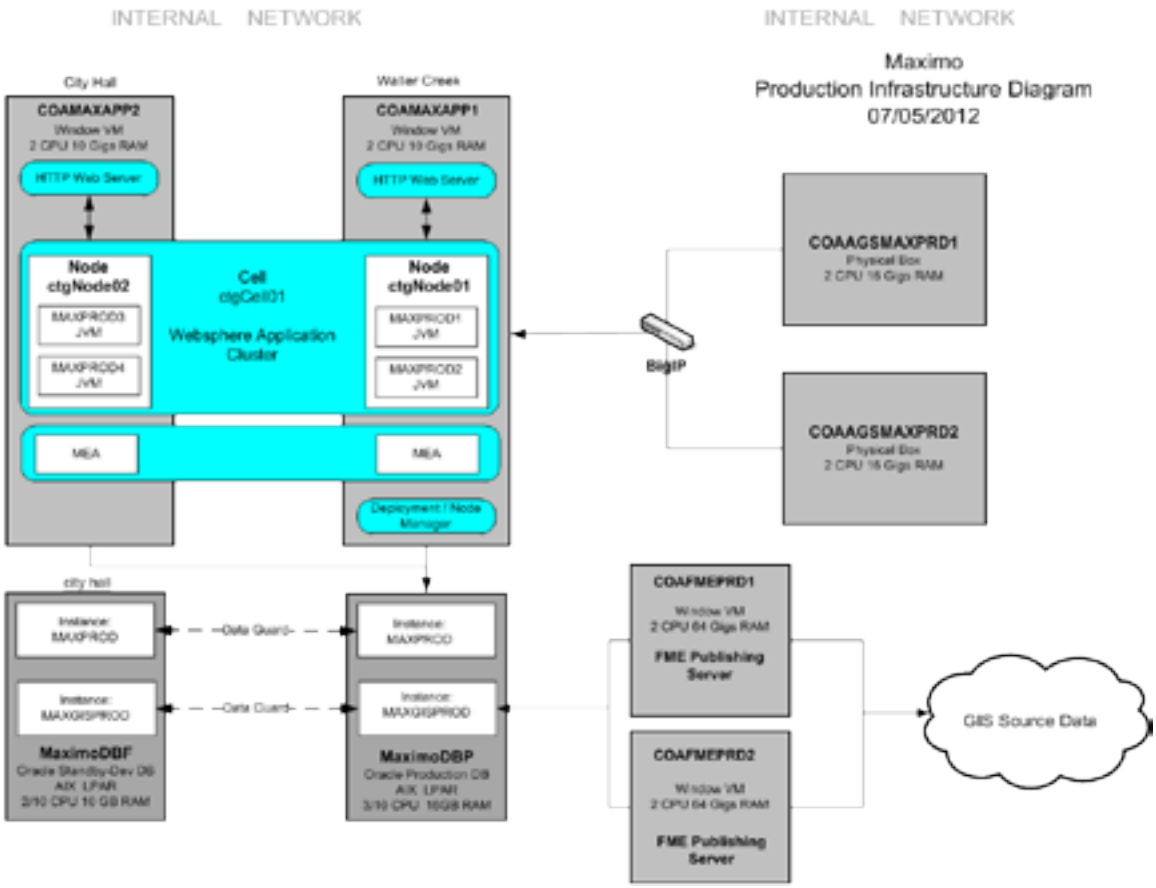
An additional ArcGIS Server service will be created from an MXD called WPD_WO_SR.MXD. The new feature classes will be symbolized and added to the Maximo Map so that users can see not only work that is going on in the area they are browsing on the Map (Open Service Requests and Open Work Orders) but also any past work that has been done in the area (Closed Service Requests and Closed Work Orders). Their visibility can be toggled in the Map Layers list.

In addition, these point feature classes will eventually be published to the City data mart to allow external spatial analysis of service request and work order histories.

Maximo to Workflow Manager

WPD GIS staff utilizes ESRI Workflow Manager for tracking work performed on the stormwater feature classes. As discussed previously, when changes are made in the field which would precipitate a corresponding change in the GIS, a child work order is generated and assigned to the GIS workgroup in Maximo. Since Workflow Manager is essentially a work tracking application similar to Maximo, WPD is attempting to create an integration between Maximo and Workflow Manager using web services. When a work order is created in Maximo and assigned to the GIS staff, the information would be transmitted to Workflow Manager to create a new GIS maintenance job. Then the GIS staff would complete the work as usual by following through the GIS maintenance job template in Workflow Manager. When the job is complete, the completion status would trigger closure of the related work order in Maximo. This process will hopefully reduce the need for GIS staff to enter Maximo and perform queries for work that needs to be done.

Appendix A: Citywide Standards for Asset Status

CODE	DESCRIPTION	INSTRUCTIONS FOR USE	INTERNAL STATUS
FUTURE	Planned or Proposed	These assets have been identified as being needed but have not yet been approved.	NOT READY
APPROVED	Approved	These assets have been identified in a master plan, approved plat or other approved design	NOT READY
UNDERCON	Under Construction	Construction has begun, but the assets are not yet ready for use	OPERATIONAL
ACTIVE	Active	These assets are operational, on stand-by, or undergoing maintenance	OPERATIONAL
TOBEREMV	To be Removed	These assets have been identified for removal and are pending decommission. Default for GIS Deletions. Removal work order may still be open.	OPERATIONAL
INACTIVE	Inactive but Still In Place	These assets are not used but were left in place and may be inspected, maintained, or reactivated	OPERATIONAL
INAC-DCM	Inactive and Decommissioned	These assets are inactive but still in place and will never again be operated, inspected or maintained	DECOMMISSIONED
REMOVED	Decommissioned and Removed	These assets are historical only, and were physically removed from their operating location and scrapped	DECOMMISSIONED
NEVERCON	Never Constructed	Though they were approved for construction, these assets were never built after approval to do so	DECOMMISSIONED

Appendix B: Maximo Production Infrastructure Diagram