Workplace and Facility Automation via the Internet

Maarten van Emmerik
Martin Waardenburg

PC and client-server based environments for workplace and facility automation are gradually being replaced by web-based environments. The advantage of a web-based environment is that everybody works on a single database and that no installation of software on user workstations is required. This not only reduces cost of ownership but also enables organizations to share data and business processes with their customers, partners and suppliers. Many vendors today claim their solutions are 'web-enabled' but often this does not go beyond offering a web interface to their existing application. A true 'web-based' system requires a fundamentally different design to address conflicting system requirements in terms of flexibility, security, ease of use and scalability.

Extended Enterprise

The increasing popularity of web-based systems is driven by economic trends towards outsourcing and globalization. As organizations need to reduce cost they tend to outsource business operations that do not belong to their core business. Instead of maintaining their own catering staff or travel agency, it makes more sense to outsource this to another company who can do this more efficiently because it *is* their core business.
Cheaper communications and airfare have led to globalization, where organizations work with suppliers who can offer best value regardless of their geographical location. Outsourcing and globalization requires efficient communications and tight integration of business processes between members of the extended enterprise such as remote employees, customers, suppliers, resellers and government agencies.

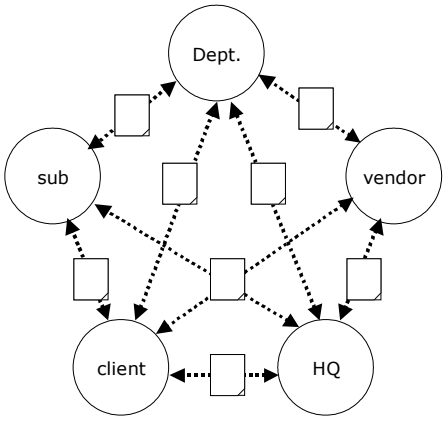

Figure 1. Data fragmentation.

Electronic messaging and email systems have made communication more efficient but they do not address the problem of data fragmentation. For example, the support department uses a request management system to track a critical customer complaint. At the same time the marketing department may be sending a sales letter to the same customer using a different information system, unaware of the current issue. Data fragmentation is even an issue within small groups, for instance when a document needs to be approved.
Five people may receive the document by email and each of recipients makes changes. It is unclear how these changes are merged, what the current state of the document is or even where the latest version can be obtained.

Sharing Data via the Internet

A first step towards solving the data fragmentation problem is to share data using shared folders, an FTP site or intranet. Instead sending documents by email they are centrally stored on a server and can be accessed via the Internet. The same approach can be used for managing data stored in a database such as names and addresses of people. Users from different geographical locations can access the contact database via the Internet and, depending on their access permissions, they can view or modify contact information.
By providing web-based access to a central database of records and documents, organizations can ensure that the data is regularly backed up and that updates and changes are immediately visible to all parties. It also protects organizations from cases where a sales manager keeps the

contact information on a PC and then all data is lost when the PC is stolen or when the sales manager is no longer with the company.
Facilitating access to information via the Internet is important but preventing unauthorized access to data is equally important. Enterprise scale web-based solutions should therefore provide a comprehensive and layered security model. Database encryption and encrypted data transmissions should be provided, as well as comprehensive tools to set access permissions for individual objects, fields and functions.

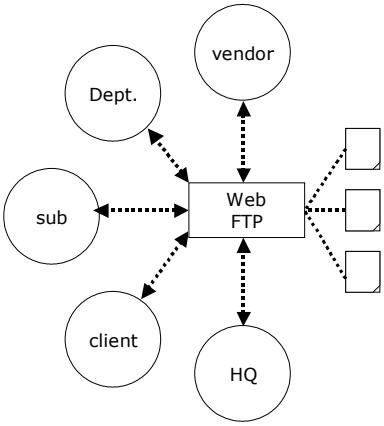

Figure 2. Consolidating information on a server and providing secure access via Internet.

Data integration

Storing data on a server and providing secure access is step one towards solving the data fragmentation issue. The next step is to create relationships between the data. Imagine an environment where two spreadsheets are stored on a secure site. One spreadsheet has a list of employees and the other spreadsheet has a list of assets. You can ensure that only certain people can access the information and that only one person can update the information at the same time. However, the data itself is still not integrated: if you look at an employee in one spreadsheet there is no link to assets in the other spreadsheet. You can not see which assets should be returned in case the employee leaves the company.
There are many ways to make links between data. Some are application specific, such as linking cells in a spreadsheet. Some are more general, such as making links from one web page to another. Probably the most efficient way to create and analyze data relationships is by using a relational database, such as Access, SQL server or Oracle. In a relational database one record representing an asset can link to another record representing an employee. Instead of using a relational database, some systems such as Zope[1] use an object database. Although object databases can be powerful, they are more difficult to integrate with other reporting tools and enterprise applications such as SAP[2] or PeopleSoft[3].

Process integration

Once data from different sources has been consolidated on a server and integrated, the next step is to define the business processes to modify the data. For example, a document may have a field 'status', which can be 'draft', 'approved' or 'rejected'. One way to change the field is to go to the document and change the value. In most cases such as direct data modification is not allowed. In case of the document there may be an official release procedure where the document needs to be approved by at least two different people. So instead of changing the field directly, a process is started where review tasks are assigned to people.

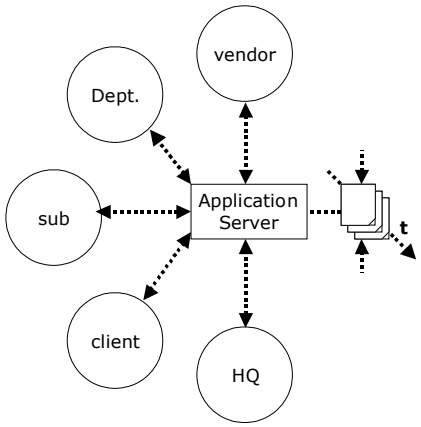

Figure 3. Integrating data and processes.

The need to implement rules and processes for updating information means that business logic needs to be stored on the server. So instead of using a 'dumb' web server that just provides access to documents and records we need a 'smart' application server that can be programmed to understand business processes.
The two main technologies to implement application servers are Microsoft .NET [4] and Java J2EE [5]. Both approaches are

suitable and have their own advantages and limitations. The .NET technology is generally considered to be easier to learn by programmers, whereas J2EE is considered better suited for large scale and multi platform environments. An important consideration for selecting an enterprise information system is the technology for implementing business logic. Some business logic in the application server is more or less static and should not be changed. This type of logic should be implemented by the vendor of the platform. For example, when a part is taken from an inventory the stock should always be decreased. Or when a document is locked for editing nobody should be able to change it.
Other elements of business logic may be very much dependent on how a specific organization works. For example, an organization may define various processes for dealing with customer requests depending on the type of requests. Based on feedback from customers these processes may be changed over time. It is therefore important that an enterprise system allows users to quickly define and modify processes without the need to request a new software release from the vendor.

Process customization

Tools for customizing business processes are typically referred to as workflow management systems. A workflow management system assigns work, passes it on and tracks its progress. It can be a standalone solution that interfaces with other products or can be embedded in an enterprise application. More information on workflow management can be obtained via the Workflow Management Coalition[6].
The convergence of Web technology, enterprise application integration (EAI) and workflow management is called Business Process Management [6].

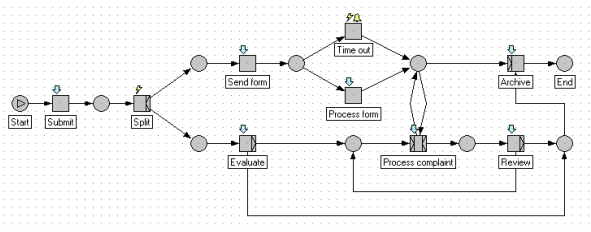

Figure 4. Workflow patterns with parallel paths and synchronization.

The most powerful and flexible workflow systems are based on Petri Nets[8], which provides a sound mathematical foundation for defining states and tasks. A state can be something like 'approved' or 'submitted'. A task is an activity that can change the state, for example a 'review' task may result in a state 'approved' or 'rejected'. Tasks and states can be connected in different ways into workflow patterns [9]. Figure 4 shows a workflow example for an incoming request illustrating some workflow patterns. The and-split pattern is used to create two parallel paths: one path represents the internal evaluation of the request and the other path starts a task to send a form to the customer to request more information. At some point in time both paths are synchronized, using the so-called milestone pattern.
Workflow management systems should be able to assign tasks to roles instead of individual people. This avoids that the workflow stalls when a specific person is not available or has left the company. For example, after an employee submits maintenance request a task is created for the role 'facility manager'. Depending on how the workflow is defined one or more persons who have been assigned to that role will receive the task.

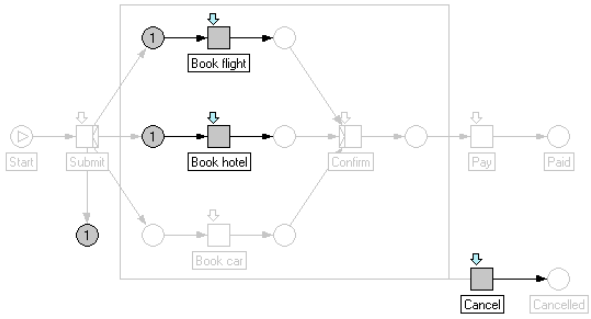

Figure 5. Graphical feedback during execution of the workflow.

Another key requirement for a workflow management system is a graphical environment for creating new workflows. This environment should preferably be web-based so that no special software needs to be installed and updates are immediately available. Graphical feedback is also important during execution of a workflow so that users see the current state and understand what happens if they press a button.

Figure 5 shows an example on how graphical feedback can be implemented. The user sees the current active tasks and can see what happens if the 'cancel' button is pressed.

Data customization

Business objects such as persons, assets or buildings typically have standard data fields. For example a 'person' may have a 'name' and 'email' address and an 'asset' may have a 'purchase date' and 'current value'. These standard data fields come with the system and are based on industry standard best practices. However, every organization has different requirements for data storage and reporting and flexibility is therefore essential.
Almost all systems allow for some customization of data fields, for instance by relabeling some special 'user data' fields. In practice this approach is restrictive because there are only a limited amount of those custom fields and they can be only of one data type, typically text. A more flexible approach can be implemented by using categories and attributes. An example of using categories and attributes for managing building related data in shown in figure 6.

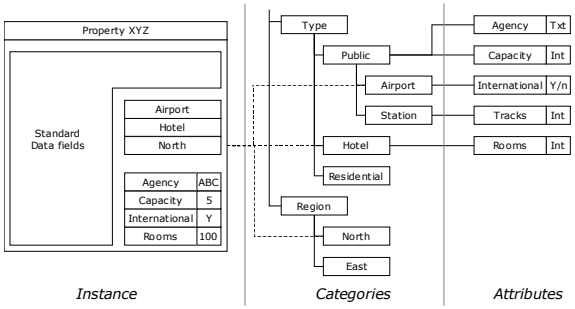

Figure 6. Data customization via categories and attributes.

When setting up the system the administrator can define custom categories and sub categories. In case of managing buildings a company may want to track the 'type' of building and a 'region'. After the categories have been defined the administrator can then define custom attributes for each category and sub category. For example, if a building is a 'public' building one may want to track the responsible 'agency' and the 'capacity' in terms of people. With a sub category 'airport' one may want to track whether the airport is domestic or international. Each attribute can be of specific type, such as text, boolean (yes/no), number (integer or floating point) or enumerated list.
When the user creates a building he or she can select one or more categories. In the example, the building is assigned to the categories 'airport', 'hotel' and region 'north'. Attributes that belong to the categories are subsequently inherited: the 'agency' and 'capacity' attributes are inherited from the category 'public' because this is the parent category of 'airport'.
The category and attribute approach can be used in various domains. When an asset belongs to the category 'computer' it may inherit an attribute 'memory' or when an organization is assigned to category 'reseller' it may inherit an attribute 'discount'. Report tools should also support filtering on categories and attributes. For example, show all buildings in category 'public' or all computers where attribute 'memory' is less than 128Mb.

Security

Web-enabled applications improve efficiency because people can access information from different locations. Equally important to facilitating data access is protecting data access from unauthorized individuals. There are many technologies to make web applications secure. There is always a trade off between ultimate security, user convenience and responsiveness of the environment.
Web-enabled applications transmit information over the public Internet. The first level of security is to encrypt the data transfer itself. The standard way to do this is by using secure socket layer (SSL). SSL is supported by most web servers and application servers. Instead of connecting via 'http', the user connects via 'https'. This secure transfer is commonly used for reservation and e-commerce sites where users need to provide credit card information.
The next level is to authorize the user, which is typically done through a unique user name and password combination. Web-based systems should automatically disable accounts after a number of failed logins and should have rules for password complexity and aging. An additional layer of access control can be implemented by defining specific IP addresses from which

users can connect. Only users from known computers or domains can have access.
The next level of access control is to define what a user can see and do after a successful login. Obviously you do not want everybody to have access to all the data and functionality. A web-based application should offer a layered security model where system administrators can set up access permissions for specific objects, fields and functions (Figure 7).
Setting up these access permissions for each individual user is very hard to set up and maintain. A common approach is to link access permissions to user groups such as 'project manager', 'public' or 'system administrator'. Subsequently a user can be assigned to one or more of these groups. It should also be possible to assign users to groups for a specific context: for example, a person may be assigned to the 'project manager' group in one project and to the 'viewer' group in another project.
The actual security settings are often defined as 'full', 'view' or 'none'.

Security Level	***Example***
Class	Only an 'inventory manager' may edit data related to a 'catalog' item.
Instance	Document 'D-007' is only visible to 'management'.
Field	An employee can change an address of an organization but not the customer number.
Function	Only project managers can 'delete' a project task.
State	When a budget has been 'approved' the amount can not be changed.

Figure 7. Layered security model.

Setting up the security settings can be a lot of work and it is essential that the system provides a convenient user interface to set up a security matrix for the given set of user groups, fields and functions. A lot of work can be saved when the system supports user group inheritance. When creating a new user group it can inherit all setting from a parent group so that only the changes need to be entered.
The implementation of the security model is often a key differentiator between web-enabled applications. This is not just with respect to the flexibility and ease of set up, but also with respect to system performance. Imagine a system with 20 user groups where a user looks at a list of 500 assets showing 10 data fields. This means that potentially 100.000 checks against the security database need to be made in order to ensure that no unauthorized data is displayed.

Meta information and Access logs

In addition to a comprehensive security model, web applications should support auditing and tools for keeping track on system access. For example, the system administrator gets a report with all the system logins from the previous day and can see that several login attempts have been made from a certain computer. The administrator can subsequently decide to block this address.
On the level of business objects such as persons or documents the system should maintain basic information on who created the object at which time, and who made the last modification at which time. The meta data on object creation can be used in workflows and security setting, for example only the creator has the rights to delete the object. Information on last modification date can be used to evaluate how up-to-date the information is. In case of a contact database this will prevent the contact list from becoming stale.
The system should also support optional access logging on the level of individual fields and functions. In case of 'assets' one may want to track all changes to the field 'current value'. In case of a document one may want to track who has viewed or downloaded the document at what time. Access logs provide useful meta data about how people are using the system. It also improves accountability because it is easy to check if someones claim that he has 'never seen' a document is true.

User Interface

A general limitation of web-based systems compared to windows-based systems is that they are less suitable for high-volume data entry. Windows systems often have key shortcuts and quick access to selection lists because everything is running on a single system. With web-based systems there is always a trade off between having small pages that download quickly versus having large pages with a lot of intelligence

for fast navigation. When evaluating a web-based system check for implementation of TAB shortcuts to move to data fields, the ability to use arrow keys to navigate through selections and keyboard shortcuts to open and close screens.
Since most users are familiar with Windows applications it helps if the graphics design is similar to applications such as Outlook or Access. The interface should include icons with tool tips, folder navigation and on-line help pages. The use of pop-up screens should be avoided since may users have installed so called pop-up blockers. Pages should have navigation help, such as a history trail, so that users do not get lost.
As organizations are becoming more internationally oriented it is important that a web-based system has built-in localization support. This includes the ability to switch languages, where not only the screen labels are translated but also messages, tool tips and help files. Localization also includes support for time zones so that dates are converted to the local time of the user. It should be possible to work with different unit measurement systems (metric, imperial), currencies, time and date formats (12/31/02 10:00 PM or 31-12-02 22:00).

Scalability and Availability

With web applications multiple users work concurrently on a server. The server should keep track of user sessions and ensure that simultaneous updates from users do not compromise the integrity of the database.
As more users come on line the server may not be able to timely process all requests so that hardware upgrades are required.
When twice the computing power is needed it may be possible to purchase a faster server but when 100 times more computing power is needed this does not work anymore. In this case a special scalable system architecture is required. Scalable web solutions enable a single application to run on a cluster of servers. The workload is spread out over multiple servers via a load balancer. Load balancing is often implemented in combination with hot fail over: when one server is down the work can be automatically directed to another server.
Running a high performance web application in a load balanced environment is not straightforward. In order to achieve high performance the application needs to store certain data in memory to avoid the overhead of querying the database every time. When one node makes an update to the database it should somehow send a message to the other nodes that they need to refresh their data stored in memory. Scalable web applications also should support the concept of database transactions. If something goes wrong while processing a complex operation the operation should either finish completely or fail completely but not leave the database in an undetermined state.
The Java J2EE environment has built in support for transactions and application clustering. However, if an application is implemented in J2EE this does not automatically mean that it does support it.

Modular Architecture

Related to scalability there is a requirement for modularity. This means one can start with using the system for one area such as managing projects, contracts or requests, and later add other modules.
Figure 8 shows an architectural diagram of a scalable web-based environment for workplace automation. On the left side are the users who connect to the system using a Web browser such as Internet explorer or Mozilla. The middle tier represents the server environment which includes some core technology that is shared between different modules. The application server connects to a database server where the information is stored.

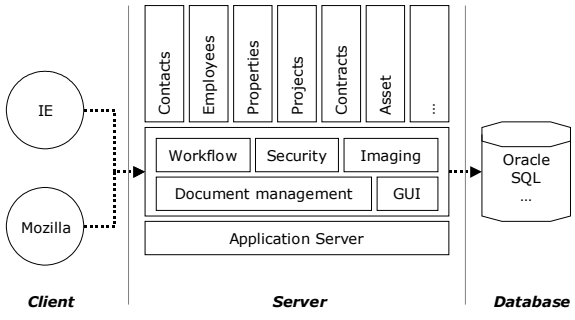

Figure 8. Modular architecture.

Enterprise applications should be able to support different databases, depending on customer requirements. These include industry standard database such as Oracle and SQL Server, but preferably also free open source databases such as PostgreSQL [10] and SapDB[11].

A workplace and facility management system should include modules for managing data and processes related to all common business operations. This includes keeping track of employee time sheets, reserving conference rooms, collaborating on projects, submitting maintenance requests and ordering office supplies. Some key requirements for these modules are discussed below.

Document Management

Almost all corporate knowledge is stored somewhere in documents such as reports, spreadsheets and budgets. Some companies make it their core business to produce documents, such as legal firms or architectural drawing offices. Yet many organizations don't go beyond setting up a shared folder structure for storing documents, or even worse, let employees manage documents on their own PC.
A reason for using a document management system is to provide better control on who can view or modify a document. Another reason is to enable quick searches based on keywords, author, date status or custom attributes so people spend less time on searching for information and do no duplicate work that has already been done. Document managements systems also help to manage revisions by ensuring that only one person at a time can make changes and by keeping track of previous versions.
Some document management systems also provide functionality for document conversion and viewing. For example, when uploading a document it is automatically converted to PDF. Users that only have 'view' permission to the document can download the PDF version but not the source document. Document conversion is also useful for facilitating viewing of documents. When uploading a floor drawing created with AutoCAD[12] the original file is converted to PDF or DWF[13] so that persons who do not own a copy of AutoCAD can view the drawing in a web browser.
Imaging is another important requirement for a web-based document management system. Large image files such as scanned drawings or contracts are not suitable for viewing in a web browser because it takes too long to download and many formats are not supported by a browser. An image server can sample the high resolution image on the server and return a web viewable format such as JPEG or GIF. The image only needs to be at screen resolution and can be 50Kb versus 5Mb for the high resolution scanned page. When the user zooms in, a subset of the image is re-sampled on the server and sent back to the browser.
Document management should be tightly integrated with all modules of the workplace automation system. It should support the ability to attach documents to business objects, such as attaching a photo and contract to an employee or attaching a drawing to a floor in a space management module.

Contacts and Employees

Contact management is a core module in a workplace automation system. In this module all organizations and persons are stored. It is important that the contact module can adequately represent hierarchical relationships between organizations, departments and persons. Hierarchical relationships can be used in workflow processes, for example if a task is not timely performed a notice is sent to the 'manager' of a person. It should also be possible to assign a person to multiple departments without having to duplicate the basic person information such as name, birth date or gender. Persons can have functions in different organizations and departments. It should be possible to enter email and phone information for each function.
The contact management module should also provide functions for defining access permissions and for automatically sending personalized email messages to people and organizations in the contact database. The contact management module can be enhanced to include CRM functionality such as track records of interactions, purchase orders, outstanding invoices and products and services.
For contacts that are also employees more detailed information can be stored. Employees are often the most valuable and expensive assets in an organization. An employee management modules helps organizations to keep track of key employee information such as work experience, qualifications, evaluations, salary history, vacation days and absence.

Request Management

Until recently most companies used paper based procedures for requesting vacation days or ordering office supplies. Today these procedures are often replaced by email messages and digital forms. A limitation of email based request management is that procedures may not be strictly enforced and that it is not easy for the requester to see the current state of the request. So people have to send reminder messages or call to find out what the status us.

Request Forms
Maintenance request
Vacation request
Document approval
Conference room reservation
New hire
Request for information
Change order
Departmental move request
Request for quotation

Figure 9. Examples of requests.

A better approach is to install an employee portal where requests can be submitted on-line and where the employee can track the request status. For each request a specific form and workflow can be designed. Department managers can quickly get an overview off all outstanding requests and can ensure that service level agreements are observed.
Web-based request management systems can be made available to other users in the extended enterprise besides employees. For example, a customer can submit a support request or order a product. These 'self help' systems are often perceived more user-friendly and effective than phone based systems since they are available 24 hours per day.

Contract Management

All organizations deal with contracts and legal agreements. These can be property lease contracts, service level agreements, employment contracts or maintenance agreements. One way to manage these documents is by archiving the paper copies in a file cabinet and keeping track of the contract list in a spreadsheet. A web-based workplace automation system can offer a more efficient approach by storing scanned versions in the document management system and providing secure access via the Internet. This way all authorized persons have immediate access to the agreements and do not need to keep their own copies or request fax copies. Figure 9 shows an initial view of the page a low resolution. When the user zooms in on an area a high resolution image of that part of the page is shown.

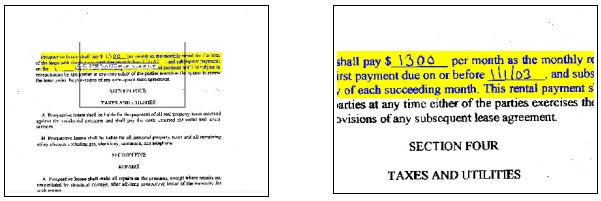

Figure 9. Linking contract clauses to scanned documents using imaging technology.

Many organizations make an abstract of each contract by summarizing the main clauses, such as for payments, extensions and renewals. Figure 9 shows how these clauses can be integrated with the original document using digital imaging. When the user clicks on a clause the corresponding page in the signed contract is shown.
A key function of a contract management system is to keep track of important dates and payment schedules. If an organization has 2000 agreements and each agreement has 10 important dates there are potentially 20000 dates that need to be managed. If somebody forgets to exercise a lease extension option it could happen that the entire department needs to move out. It should be possible to link contracts to other objects in the system such as employees, assets or buildings. If an organization has signed a maintenance contract all work orders can be booked against the budget agreed in the contract.

Property and Space Management

Office space is often the highest expense after employee salaries. A property management module can help to reduce cost by ensuring that space is used effectively. It can also help to keep cost under control by keeping track of maintenance agreements, assets, licenses and compliance documents.
A property and space management module should have the ability to link people, areas and assets to elements in a floor plan. For

example, if the user moves the pointer over a drawing some feedback about the objects is shown. By clicking at an element the system can then show the associated information that is stored in the database or navigate to another web page with more details.

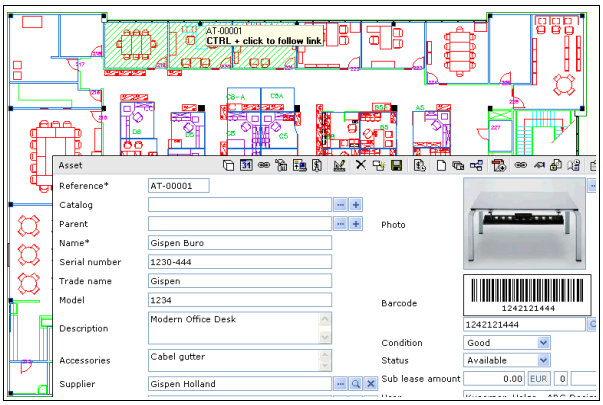

Figure 9. Linking elements in a drawing to business objects in the workplace automation system.

A system for specifying categories and attributes as discussed earlier is essential for space management. Categories can be used to classify areas as common, circulation, usable or rentable. This classification is then used to calculate building occupancy, find available space or to allocate expenses to departments that occupy the space. Custom attributes can be used to store additional information, for example if a space is of category 'conference' then it will inherit attributes such as 'seating arrangement', 'wireless internet' and 'video projector'. When making a reservation, users can then search for a conference room with 'classroom arrangement' and a 'video projector'.

Assets and Inventory

Assets may be owned or leased but in both cases they need to be carefully tracked. Many organizations only track assets in the financial system to calculate depreciation and current value for the balance sheet. A workplace and facility automation system typically tracks much more information such as where the asset is located, who is using it and its current condition. Assets can be linked to areas and floor plans, to employees who are using them and to documents such as user manuals and lease agreements.

Inventory are items that are stored somewhere in a storage location such as a warehouse. Inventory items are often consumable products that are not tracked individually, however they can also be durable assets such as furniture or equipment that can be assigned to people or organizations. Managing an inventory implies keeping track of incoming and outgoing transactions, stock levels and purchase orders. The advantage of a web based system is that inventory levels can easily be checked via the Internet and that suppliers can manage inventory levels of their products on line.
Generic information about assets and inventory items can be stored in a catalog. For example, the catalog can have an entry 'Dell 5100 notebook' or an entry '40W light bulb'. At this level generic information such as photos, manuals, weight and dimensions can be stored. If the item represents a durable asset, in case of a computer, it should be easy to query how many assets of this type are used in the organization. If the items is a consumable, it can show the stock levels in the various inventories.

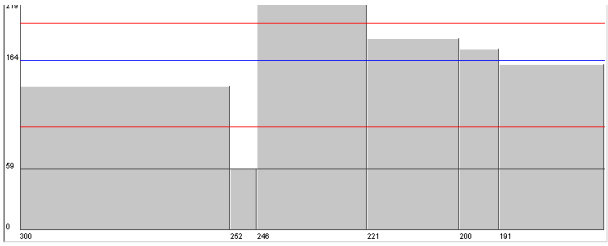

Figure 10. Graphical report showing inventory levels.

As with all modules, it should be easy to run reports and save these reports as a PDF file, or to import the results into Excel for further processing. The reporting tools should also include capabilities for drawing charts such as shown in figure 10.

Project Management

Projects are activities that run during a certain period such as building a new office, moving to a new floor or organizing a conference. Project management has been one of the first areas where web-based collaboration tools have been used because they provide an environment where people from different organizations can work together.
A project management module should provide functionality for setting up a project team, collaborating on documents, tracking

requests and issues, messaging, keeping track of the project schedule and assigning project tasks. Team members are assigned to one or more user groups that define their access permissions and workflow tasks.
A project management module should also assist in keeping track of the financial aspects of a project. This includes setting up a budget and managing the budget by keeping track of what amounts have been committed and paid. The committed and paid amounts are obtained from purchase orders and invoices that have been submitted against the project.

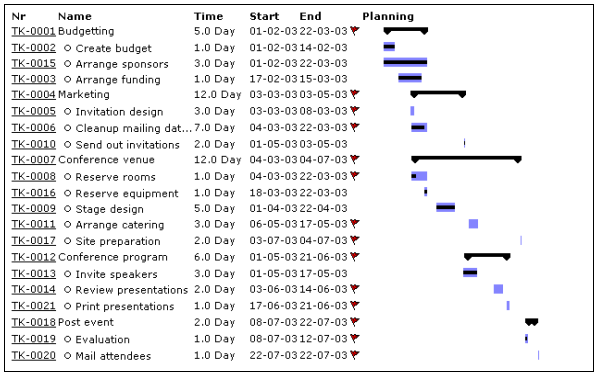

Figure 11. Project schedule accessed via browser.

Time sheets are essential for projects that are billed on an hourly rate. Even when a project is fixed price it is advisable to require time reporting by team members to keep track of the internal cost. Entries in the time sheet should be linked to project tasks so that the project manager can compare the estimated time and cost with the actual time and cost. This information can be used for future cost and time estimations.

Integration

The ability to integrate with existing data and applications is an important requirement since most company already have data in some kind of format. A key consideration with data integration is to decide where the master data resides. An organization may use an LDAP server [15] as the master source for person information which then needs to be synchronized with the contact module. Another organization may decide that the contact module is the master source and from here employees can download information in their personal contact management software.
There are various approaches for data and application integration. On the lowest level data can be integrated on the database level, for instance by allowing other applications to share the database used for the enterprise system. This approach is suitable for reading data but not for modifying data because business logic of the application is by-passed. Adding or changing information requires a special interface in the workplace automation system. This interface ensures that data is validated and that associated data is updated.
The actual data feed can be supplied in various ways. Most applications can export data in a comma separated format (CSV). This file can then be uploaded via a web interface and columns can be linked to data fields and attributes in the workplace automation system. The import module updates or creates business objects and ensures that certain update rules and business logic are enforced.
The most flexible approach towards application integration is by using a web services Application Programming Interface (API). Third-party applications can query and update information through remote procedure calls based on XML/SOAP [16]. The advantage of this approach is that SOAP works over a standard HTTP(s) connection and that it is supported by many programming environments such as J2EE and .NET.

Application Service Provider

A final consideration for selecting a workplace and facility automation system is whether is should be provided as software or as a service. Several software vendors now offer their systems as an Application Service Provider (ASP). Customers can purchase a subscription instead of buying the software.
In the ASP model the system is hosted on servers that are operated by the software vendor. Customers do not need to invest in hardware, software and personnel. No large upfront investment is needed, which reduces financial risk and reduces capital expenditure. In many cases an ASP model provides a higher availability than running software in house because the system is hosted in an environment with 24hr monitoring, backup power supply and redundant Internet connectivity.
The ASP model is especially suited for smaller companies that need enterprise

level functionality but do not have the budget or skills to maintain such an environment. It is also very suited in cases where different companies work together as a conglomerate or alliance, either on a project basis or permanent basis. By using an ASP the members may feel more comfortable when a neutral third party hosts the data instead of the system being controlled by one of its members. Sometimes ASP vendors offer hybrid models, where the product can be used as an ASP during a pilot phase. Once this phase has been completed a dedicated environment is set up for the customer, either at the customer side or at an external service provider selected by the customer.

Authors

Maarten van Emmerik and Martin Waardenburg each have over 15 years experience in development of enterprise systems for computer-aided design and business process management. They founded Axxerion to develop a new generation workplace and facility automation software based on the concepts outlined in this paper.

References

[1] Zope, Application server for content management (www.zope.org).
[2] SAP, Provider or ERP solutions (/www.sap.com).
[3] Oracle, Enterprise database and application provider (www.oracle.com).
[4] Microsoft .NET, Software technology from Microsoft (www.microsoft.com/net).
[5] Java J2EE, Software technology developed by Sun (java.sun.com).
[6] Workflow Management Coalition, Non-profit organization (www.wfmc.org).
[7] Business Process Management Initiative (BPMI), Industry Initiative (www.bpmi.org)
[8] Petri Nets World, Online directory , (http://www.daimi.au.dk/PetriNets/)
[9] Workflowpatterns, Repository of patterns (www.workflowpatterns.com)
[10] PostgreSQL, Open Source Database, (www.postgresql.org)
[11] SapDB, Open Source Database, (www.sapdb.org)
[12] AutoCAD, Computer Aided Design software (www.autodesk.com).
[13] DWF, File format for viewing drawings via Internet (www.autodesk.com)
[14] Axxerion, Enterprise workflow management (www.axxerion.com).
[15] LDAP, Lightweight Directory Access Protocol (www.ldap.org)
[16] SOAP, Simple Object Access Protocol (www.w3.org).