Enterprise Content Management with Interstage Contentbiz

● Matthias Bandorf ● Tsuneichi Yoshizawa ● Yuji Takada
● Guenter Merbeth

(Manuscript received January 5, 2004)

In today's competitive business climate, enterprises must be able to mange information effectively and efficiently. Enterprise content management systems are emerging as key tools for meeting these requirements. They can manage data throughout an entire enterprise as important information assets. Interstage Contentbiz is an enterprise content management product from Fujitsu that manages structured and unstructured data as content and can even integrate disparate, distributed data resources.

1. Introduction

Information is one of the most important assets of enterprises. The method used to store, prepare, and provide information plays an important role in today's competitive business climate.

Enterprises have to deal with quite a lot of different kinds of information, for example, balance sheets, stock data, invoices, reports, drawings, and even audio and video data. Very often, this data is classified as structured or unstructured data. Structured data is typically managed by relational databases. Unstructured data is mostly found in office documents and is stored in file systems, document management systems, and repositories.

Traditionally, information processing focused on the management of structured data in relational databases. However, only around 20% of all data stored in computers is managed by a relational database management system (RDBMS). The other data is mostly managed incorrectly. It is stored in a variety of systems, sometimes even on local disks. The consequence is very often that data cannot be found, outdated information is used, data is kept redundantly in several copies, changes are not managed properly, and so on.

When Internet technology became popular, data for Internet/intranet (Web) sites had to be managed and a new category of products called Web Content Management systems became important. These systems manage information to be published over the Web (called content in this context).

During the last few years, Web technology has become the preferred platform for IT systems to improve business processes. The term "e-business" has often been used for this kind of system. It is obvious that these systems must have access to information stored in the IT systems of enterprises. For many business processes, a larger set of various types of information must be made available. It is also obvious that Web Content Management systems do not have the functionality or power to fulfill the requirements of e-business applications. Products that offer functionality to make information from different sources available for e-business applications are called Enterprise Application Integration (EAI) products. These products have adapters to get data from several sources and also provide

workflows to combine this data for applications. However, EAI products have a disadvantage: the integration of an application with the back-end sources is very often hard-wired. This disadvantage can be overcome by inserting a product between applications and the back-end adapters that provides flexibility in information provisioning and also provides options that add value to combined information from back-end systems.

A new category of products called Enterprise Content Management (ECM) systems are now available that meet these requirements. These products integrate information (content) from different sources, form it into a collection (compound content) and provide it to users and applications, and add value to the represented information (e.g., metadata, dependencies, and categories). Information managed by ECM systems can be stored directly in the ECM created in the system or imported from another system. However, in most cases the original data remains in the original system, a relational database, a packaged solution such as SAP, a document management system, or a mail system. In other cases, the ECM system keeps only a reference to this data and adds additional information to it. Therefore, content integration is a very important aspect for ECM systems. ECM systems must be able to integrate structured data as well as unstructured data and also combine it into compound content with added meaning.

Fujitsu's Interstage Contentbiz, also known as Interstage Content Integrator, is an ECM system that focuses on content integration. It offers state-of-the-art functionality and in some areas has market-leading features.

2. Enterprise Content Management

Content management is a functionality to help users effectively create, utilize, and share content. For this functionality, a content management system provides users with various functions such as lifecycle management, metadata management, searching, and authoring.

Because content management is a functionality required in a wide variety of situations and purposes, various types of content management systems are emerging in the market from different viewpoints and contexts. These can be summarized as follows:

1) Document management systems

Document management systems mainly manage the lifecycles of content in a sophisticated way so that users can work on and share content effectively between teams. Because of their rich and sensitive treatment of content, they can be used in important areas such as record management. Documentum, FileNet, and Stellent are typical vendors offering document management systems.

2) Web content management systems

Web content management systems mainly support activities related to Website management. They support content authoring and content publishing on Websites with a template function and transactional content delivery. Vignette and Interwoven are typical vendors of Web content management systems.

3) Enterprise information portals

Enterprise information portals are systems for showing integrated and personalized views of various distributed content resources in Web browsers. To create sophisticated views, they often manage content so they can access and render it effectively. For this purpose, some portal products have a content management function. Plumtree, Microsoft, and SAP provide this type of portal product.

4) Search and classification products

Effective identification of content in a large volume of content is essential in content management. The most effective way to achieve this is to perform a full-text search and classify the content. Therefore, some vendors of search and classification products, for example, Verity, Autonomy, and Hummingbird, also position their products as content management systems.

5) Enterprise information integration products

These products integrate multiple databases and repositories. They provide users with a seamless and unified access to different types of databases and repositories. To achieve this, they typically manage the metadata of content. BEA and IBM offer this type of product.

ECM is a new generation of content management. It has functions for the traditional content management described above; however, its most important features are its content infrastructure and content integration.

1) Content infrastructure

An ECM system is a middleware platform for content-oriented solutions. Solution applications that use content can be effectively and efficiently developed and deployed without a large system-integration effort.

2) Content integration

An ECM system integrates disparate content resources distributed over a network so users can access content seamlessly, wherever they are stored and whatever format they are in.

Some examples of content-oriented solutions are customer relationship management, knowledge management, digital asset management, and the enterprise information portal. These solutions are introduced with their own content repositories. However, these repositories are independent silos and cannot be accessed by other solutions. Therefore, better functional solutions are required so that more content stores become available to these solutions. Therefore, solutions require a content integration infrastructure. Then, all of the content management functions can be consolidated into the infrastructure, new solutions can be developed, and existing solutions can be rebuilt on the ECM system.

Figure 1 shows the framework of ECM. The content integration component integrates disparate content stores distributed over a network. Content should remain in these repositories in the format of each repository. The content integration component should be able to access them on demand. This enables content to be kept fresh and at the same time enables each application to manage content in its own way.

The content management component realizes all the functions provided by traditional content management systems, for example, lifecycle management and authoring. Because content remains in external content stores, the content management should be based on metadata.

The content distribution component provides content-oriented solutions with various sophisticated access methods. Search-based and

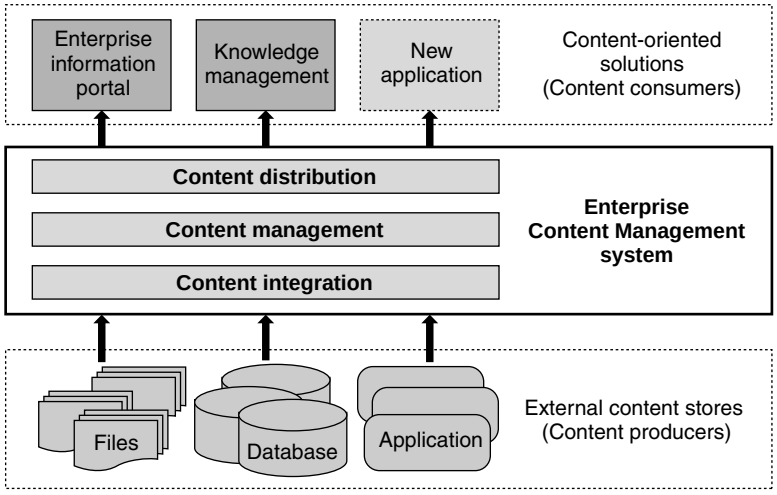

Figure 1
Enterprise Content Management framework.

category-based accesses are easy ways to identify content. Personalization and recommendation are sophisticated ways to present content to users. Periodic, transactional delivery of content is required to publish and syndicate it.

Enterprises require ECM to stay competitive in the market. They are currently introducing content management systems to enable users to effectively use information. This introduction is mainly based on small sub-organizations. However, since an enterprise's IT infrastructure is enhanced by network connection, it becomes possible for an enterprise to empower users with more information by integrating their content repositories and business applications. Such an integration becomes more important in mergers and acquisitions: information in systems from one party should be available to another party. ECM systems make this possible in an effective way.

An increasing number of content management products support content integration. However, most of them are not ECM systems yet. This is because they still focus on their own applications and are still difficult to use as a content infrastructure.

3. Integration scenarios

An ECM system can have many valuable uses in an enterprise. It can be seen as the underlying system for building a content infrastructure for the entire enterprise that can be used in many different scenarios. This section describes some typical integration scenarios for an ECM.

3.1 Access to enterprise information from business applications

Many applications in an enterprise need to communicate with other applications. A very popular example is a customer relationship management (CRM) application. Usually, when a CRM application is introduced, a huge part of the budget and time is spent on integrating it with other applications. In this scenario, a CRM application has to be integrated with an ERP system, a knowledge database, and a product catalog:

1) The ERP system must be accessed to learn whether a customer's bill has been paid.
2) The knowledge database provides information that helps call center agents solve problems.
3) The product catalogue is used to obtain detailed information about products purchased by clients and offer clients upgrade options.

In a typical project, there are two options:

1) Build specific adapters for each system to be integrated.
2) Implement an EAI infrastructure.

An EAI typically makes sense when many applications have to communicate with each other. However, experience has shown that EAI projects typically take a long time to implement and are expensive. Therefore, the situation has to be evaluated carefully to see if it is worth executing such a project.

When an ECM system is available, there is a third option: making the required content from external systems available in the ECM system. Therefore, the content can be accessed easily from the CRM system, either through the Java API or through Web Services.

Of course, an ECM system cannot always replace an EAI project. However, content integration is sufficient in many scenarios. Also, it is much easier to implement a content integration project than an EAI project because:

1) Only access to the content, for example, an SQL query for a relational database, has to be provided.
2) Applications can access content in external systems transparently; that is, without needing to know the methods used to access it.
3) The integration project can be implemented step-by-step. It is not necessary to model all the content to be integrated before starting the project. Instead, the model can be extended over time, even without interrupting running applications. Therefore, the project

can be implemented in small steps that provide an immediate return on investment.

3.2 Access to enterprise information from portals

When building an enterprise information portal as the central platform for accessing content throughout an enterprise, the first thing to consider is the portal software, which is needed to implement such a project. The content in each external system can be accessed through the proprietary methods of the external systems. Content from a relational database can be accessed by writing SQL queries, content from a CRM system can be accessed through a vendor-specific Java API, and content from an ERP system can be accessed through a vendor-specific, proprietary API.

However, this approach has several disadvantages:

1) Programming overhead

If the same content is needed in different environments (e.g., in other applications such as CRM) the programming to access this content has to be done twice.

2) Maintenance is difficult

If the content changes or slightly modified information is needed, a programmer is needed who understands both worlds; that is, the external system that delivers the content and the portal environment.

3) Modeling information is part of the implementation

If content of different external systems logically belongs together, for example, customer information from an ERP and a CRM system, this has to be programmed in the application logic.

An ECM system overcomes these disadvantages in three ways:

1) Reduced programming overhead

Different applications can reuse content that is already configured for use through the ECM system.

2) Simplified maintenance

When content changes or additional information is needed, these changes only require a change in the model configuration instead of changes in many application programs.

3) Modeling information in the ECM system

The semantic relationships are stored in the system. Therefore, the logic for accessing content and detecting connections between the content of different business applications is provided by the ECM system. The application programmer can access all this information in a program by reading the content and its relationships through a single API. This API allows the programmer to access all external content and obtain data returned in a unified format such as XML without needing any knowledge of external systems.

3.3 Authoring process support

An ECM system provides access to content in other business applications together with metadata information (i.e., attributes and relationships between content). Additionally, content can be stored internally in the system repository. This provides great flexibility. Content can be:

1) Referenced from the ECM system

The content remains in the external system and is referenced from the ECM system.

2) Migrated to the ECM system

The content is moved from another application to the ECM system and is stored only in the system repository.

3) Replicated in the ECM system

The content remains in the business application and is also stored in the ECM system repository. This scenario makes sense in certain situations, for example, when a business application is running outside the intranet.

4) Created in the ECM system

Content is created directly in the ECM system repository.

In a typical authoring scenario, authors usually want to work in a familiar environment, for example, Windows Explorer and MS-Office, without using additional tools. On the other hand, documents must follow a defined process. Usually,

this is done using additional document management tools that have to be obtained from authors. The disadvantage is that authors must learn how to use additional applications. This becomes even more difficult if authors are frequently changed because they are external employees. It would be desirable to have a system in place that ensures the defined process is followed without changing the environment of an author.

Many ECM systems support Web Distributed Authoring and Versioning (WebDAV), which enables authoring of content over the Web. Many commercial applications already support WebDAV (Windows Explorer, MS-Office-Suite, Adobe GoLive, Macromedia Dreamweaver, etc.). Therefore, an author can work in a standard environment without additional tools. This solution combines the best of both approaches:

1) Authors can simply use a familiar environment without making changes.
2) The actions of authors are automatically tracked from the ECM system:

- New versions are created automatically if required by the process.
- Proper support of the defined authoring process is guaranteed without the need for additional tools.

4. Interstage Contentbiz

Interstage Contentbiz (hereafter Contentbiz) is an ECM product of the Fujitsu Interstage family. Together with other products of the Interstage family, Contentbiz makes it possible to provide customers with highly content-oriented solutions that effectively and efficiently realize the usage scenarios of ECM. This section gives an overview of Contentbiz's ECM features and architecture.

4.1 Design principles

Contentbiz was designed with the following general design principles:

1) Open architecture

Contentbiz must fit into the existing IT infrastructure, because many companies have IT systems already in place for functionalities such as full-text searching and user authentication. When they introduce a new software system, they do not want to buy this functionality again, because of the software, administration, and training costs. Therefore, a new software system must be able to use the existing infrastructure. This is possible only with an open architecture.

2) Flexible data model

There is no single data model that fits all customer needs. Therefore, a data model for an open system such as Contentbiz must be flexible and highly adaptable to specific customer requirements.

3) Customizable functionality

Functionality delivered from basic services of Contentbiz must be configurable. The configurability greatly depends on the underlying data model as well as the quality of the input data.

4) Automization of work

ECM systems are intended for scenarios in which a large number of contents are handled. Therefore, the steps needed to provide access to new content in Contentbiz should be done automatically as much as possible.

5) Extendibility

Contentbiz provides functionality out-of-the-box. However, customers may have specific requirements. Therefore, the architecture must be open so that additional external systems can be integrated into Contentbiz.

6) Support of standards

Contentbiz supports standards in several areas. WebDAV will be supported to enable access to content through WebDAV-enabled clients such as the MS-Office-Suite. The Java Authentication and Authorization Service (JAAS) is supported for authentication system integration. The adapters are J2EE Connector Architecture (JCA) compliant so that JCA-compliant systems can easily be integrated.

These basic design principles make Contentbiz a very sophisticated ECM product. For

example, it provides functionality for full-text searches as well as for searches via meta-information. Any full-text search engine can be easily integrated into Contentbiz. The meta-information is stored in Contentbiz with a customer's data model. Since the data model is customizable, it must be possible to configure which metadata is used for searches. Users should be able to configure the data model to automatically extract the metadata for a newly registered content.

4.2 Features and functions

Contentbiz has unique features and functions to meet ECM requirements. These differentiate Contentbiz from other content management systems, especially from other ECM systems.

4.2.1 Modeling and organizing content

Contentbiz enables very rich content modeling. The model can be modified even in runtime, which enables very flexible and sophisticated content management.

In Contentbiz, information about content is represented by content items. Content items can represent external content (i.e., content stored in external content stores and applications) or internal content (i.e., content stored in the Contentbiz repository).

Content items are classified by content types and are described by attributes and relationships as shown in **Figure 2**.

Relationships represent the semantic relations and dependencies between the content items in a repository. Contentbiz uses in-built hierarchical relationships to build a hierarchy of content items. When a new content item is created in a repository, it is automatically connected to a parent item by such a relationship and is therefore made part of the hierarchy. The hierarchical structuring of information makes content easier to manage. This is a prerequisite to avoid redundancy, because additional non-hierarchical relationships turn simple content management into sophisticated intelligent content management. Non-hierarchical relationships between content items carry specific meanings and are therefore referred to as semantic relationships. They can be used to build network-like content structures and are the basis for navigating through the content items in the repository.

Attributes hold additional information about the locations of content and are a means of storing additional information about content.

Using relationships and attributes, comprehensive information can be defined for content. By modeling and defining the appropriate information, the various information resources can be linked together to form a powerful content

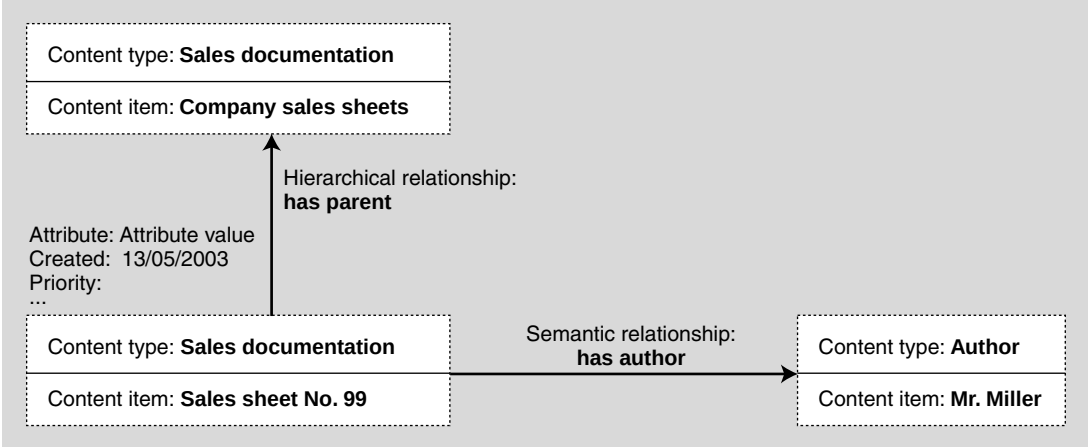

Figure 2
Basic content model of Contentbiz.

management system.

Depending on their usage, Contentbiz distinguishes between three types of content items: compound content items, simple structured content items, and atomic content items. Compound content items are nodes in the item hierarchy that act as control units. They control the access policy and the versioning technique for their subordinate items in the hierarchy. For example, the permissions set for a compound content item apply to the compound content item itself and to the content items in its substructure. Simple structured content items are nodes in the item hierarchy, which can represent actual business content. They are used, for example, for a folder containing files. Atomic content items are the leaves in the item hierarchy, which cannot have a substructure. They can represent actual business content and are used for example, for a file stored in a folder.

For classifying and organizing content with common properties, Contentbiz provides content areas. Content areas group content in the content item hierarchy, usually by organizational aspects. Content areas are compound content items at the level immediately below the root, so they define the access policy and versioning technique for their subordinate items.

4.2.2 Role-based access control model

Access control to content in Contentbiz is based on the user roles, the access rights, and the repository access levels. This access control model controls not only the content access but also the content management.

Each end user in Contentbiz is assigned at least one role that defines which content the user can access and which operations the user can execute on that content. There are three types of user roles:

1) Providers

Users with a Provider role can create new content items in the repository and request approval for content items from users with the Manager role. Providers can also add and delete semantic relationships between content items, modify attribute values, and create new versions of content items and delete versions.

2) Managers

Users with a Manager role are responsible for publishing content created by content providers. They are responsible for publishing content by approving it and rejecting content for which approval has been denied. Like Providers, they can add or delete semantic relationships between content items and modify attribute values.

3) Public

Users with a Public role can use the navigation functions that are provided for published content in Contentbiz. They can browse content, view metadata that is stored for content, and view the actual business content (e.g., view a PDF file by using Adobe Acrobat Reader). Users with the Public role can neither create nor approve content.

For users to carry out a specific operation on a content item, they not only need an appropriate role, but also appropriate access rights. Contentbiz distinguishes between “owner”, “write”, and “read” access rights.

The repository in Contentbiz is divided into three access levels. An access level is a virtual area within the repository containing a subset of content items. An access level determines the visibility and accessibility of content items and their versions according to the user's role type.

Corresponding to the user role types, each repository has three levels. The Public level is for users with a Public role. These users cannot access or view content below the Public level. The Manager level is for users with a Manager role. Managers can see all items on the Manager and Public levels. However, they cannot see content stored on the Provider level, which is for users with a Provider role. Providers can see all items stored on all three levels.

4.2.3 Content access

Contentbiz supports standard interfaces for accessing content from applications. Content can be published for the Web or integrated into enterprise information portals such as Fujitsu's Interstage Portalworks. Application programmers can implement applications that allow end users to carry out tasks such as displaying content details in a Web browser, displaying actual content by using Microsoft Word, and customizing the content display for building individual portal views.

Contentbiz provides two methods of content access. Content navigation is a function for navigating through content that has been published in Contentbiz. Content navigation primarily works on the basis of semantic relationships defined between content items. The semantic relationships are either defined manually by the users or administrators, or automatically based on the Contentbiz metadata extraction features.

Application programmers can implement end user applications that allow users to search for content. Contentbiz supports full-text searches, similarity-based searches, and metadata searches. These searches target external as well as internal content.

4.2.4 Content management

The content management functions of Contentbiz are used to manage information about content as well as semantic relationships between content in the repository. Contentbiz users can use these functions to create, update, and delete content as well as semantic relationships. The tasks related to these operations can be highly automated.

Contentbiz ensures that changes to external and internal content are automatically detected and appropriately handled by services. For example, when a content is created, a service called Metadata Extractor is invoked and automatically derives information stored in the actual business content (e.g., creation date, author, and size) and makes it available as attribute values and relationships for the relevant content items in the repository.

Contentbiz's main content-management function is to manage content lifecycles and versioning. Each content item in Contentbiz is versionable and has a predefined lifecycle. This lifecycle describes and controls the evolution of the item. A lifecycle starts when a content (i.e., an item) is created and ends when the item is no longer available for use.

The Contentbiz lifecycle establishes the states that item versions can have, the sequence in which these states are arrived at, which functions are executed at transitions from one state to the next, and whether a new version of an item is created. The state of a content item version determines the access level where it is located and therefore the operations that can be executed on them and the user roles required to execute them.

Contentbiz allows two different versioning techniques that are defined by using different types of compound content items. The versioning technique being used applies to all content items in the compound.

In an open compound, new versions of content items can be created and published that are independent of other items in the compound without versioning the compound content item itself. This means that the lifecycle of an item in an open compound is independent of the lifecycle of other content items in the compound and the compound item itself.

Usually, open compounds are used when information does not require configuration management. For example, in marketing it might be neccessary to update and publish a marketing brochure independent of other marketing material.

In a closed compound, a new version of the closed compound content item has to be created before new versions of its subordinate items can be published. In other words, the lifecycle of a content item in a closed compound is controlled by the closed compound content item.

Typically, closed compounds are used in scenarios that require configuration management or map structures in which the individual items of these structures are interrelated. For example, in product development, it might be neccessary to create new versions of all the components at the same time and not allow individual component versions.

4.3 Architecture and components

Contentbiz is highly componentized, as shown in **Figure 3**.

These components are categorized into three functional blocks: the Contentbiz core, the Agent Server and agents, and the application programing functionality. The Contentbiz core contains the components for operating and managing the overall Contentbiz system. The agents are the Contentbiz components that enable access to business content that is stored and managed in systems outside of Contentbiz.

The Repository Application Programming Interface (RAPI) provides Java classes and methods for implementing end user applications on top of Contentbiz. The RAPI is the interface of the Contentbiz services. The provided services enable application designers and programmers to implement the necessary functions for working with the content items in the repository. They also support the design of complex applications on top of Contentbiz as well as the integration of Contentbiz with other products.

In addition to the RAPI, a TAG library is provided that offers custom tags for JavaServer Pages (JSP) to facilitate the use of the RAPI. The custom tags focus on content access functionality.

4.3.1 Repository

The Contentbiz Repository stores and manages information about content as well as the underlying data model; it also stores internal content. The repository provides uniform access to content collected from different systems while maintaining the independence of each system.

The repository is the logical unit used for storing content. Each repository in Contentbiz is based on a so-called datastore, which forms the physical unit (Fujitsu Enabler). Enabler Repository is a proven technology that is used in 3000 sites all over the world with large applications. It is based on an object/relationship/attribute model and provides functionality such as version control and management of complex configurations. These concepts are implemented in the core database engine, which allows efficient handling of these complex operations compared to solutions in which this functionality is built on top in an additional layer.

4.3.2 Services

Services provide all the functionality for accessing and manipulating content in Contentbiz as well as for creating and maintaining the underlying data model and the users and roles. The RAPI is the interface for the Contentbiz services.

The Search service supports a combined full-text search and searches content metadata stored in the repository, for example, the author of a document or the creation date. The search covers both internal and external content.

The Categorization service is used for implementing and integrating taxonomies with categories. The corresponding content of a specific category can then be browsed and retrieved.

The Authentication service is used for verifying user information (IDs, passwords, etc.) against an external authentication system. The Authentication service is based on the JAAS. The following systems can be configured: Lightweight Directory Access Protocol (LDAP) and LDAP with Secure Sockets Layer (SSL), Active Directory and Active Directory with SSL, Microsoft Windows NT LAN Manager (NTLM), and Network Information Service (NIS).

The Content Collector forms the interface between the repository and the agents. It registers external content in the repository and collects content from resources outside Contentbiz. When

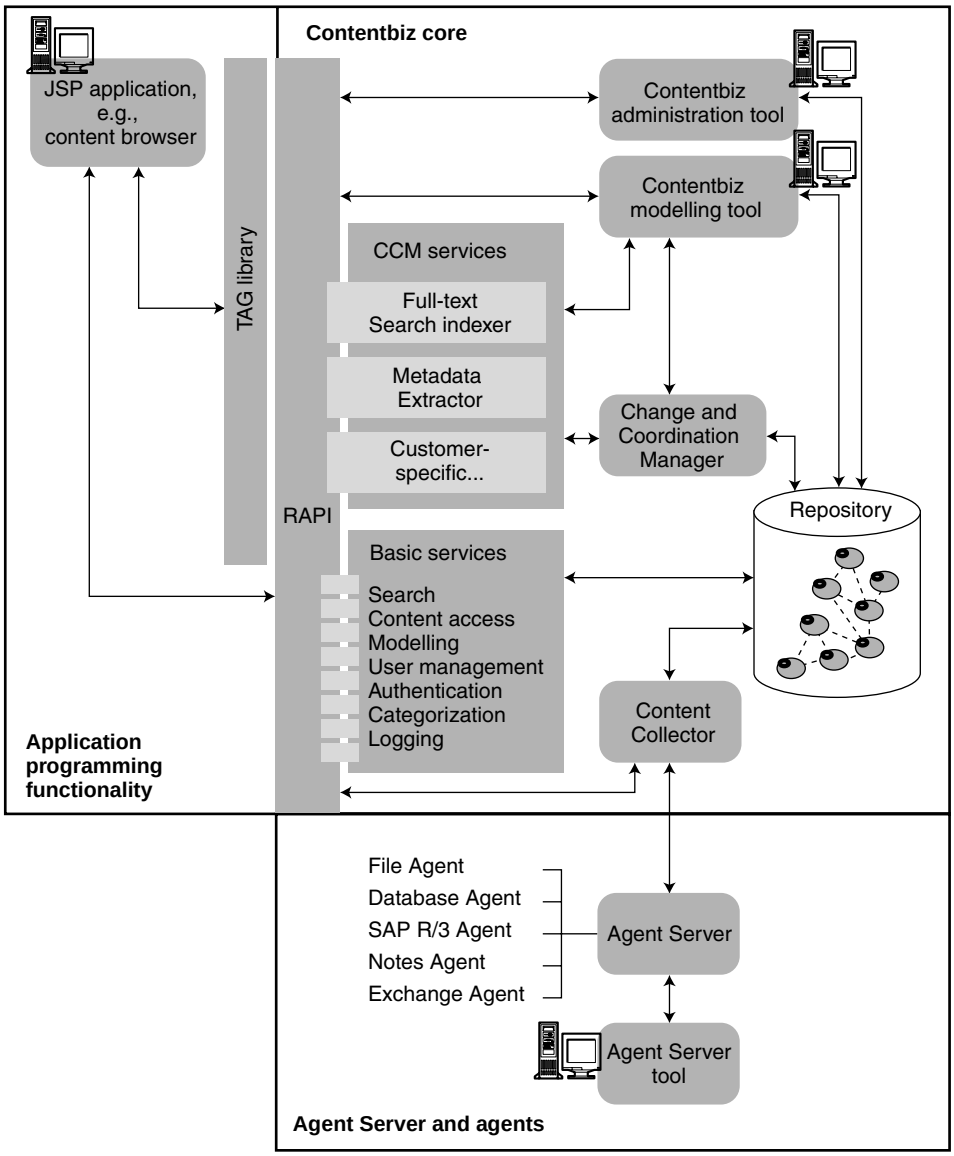

Figure 3
Contentbiz architecture and components.

a user accesses an external content, the Content Collector retrieves it from the agent and returns it based on the specified conditions. It also caches external content in the repository.

A service is invoked by a calling component, for example, by a content browser or another service, and then the requested action is performed. Typically, a component uses a service provided by another component. For example, the Content Access service is used by the Search service and most of the other services. On the other hand, the Content Access service uses the Content Collector to access external content.

In addition to these basic services, Contentbiz includes special services that act under the control of the Change and Coordination Manager (CCM).

The CCM ensures that changes to content are handled appropriately by special services such as the Metadata Extractor and the Full-Text Search Indexer. The CCM triggers the services and coordinates their tasks, therby controlling their execution and sequence. The architecture

of the CCM is flexible, so further special services can be added by programming.

The Metadata Extractor is a service that automatically derives information stored in external or internal content (e.g., the creation date, author, and size) and makes it available as attribute values and relationships for the relevant content items in the Contentbiz repository. The Contentbiz system designer defines which metadata is to be retrieved and how it is to be stored.

The Full-text Search Indexer is a service that integrates a third-party search engine into Contentbiz, thereby enabling full-text searches for content. This service triggers the creation and update of the full-text search index based on specified conditions.

4.3.3 Agents

On request from Contentbiz end users, the Agent Server invokes an agent for searching an external system based on predefined conditions. It collects the business content that matches the conditions and returns it. The agent uses XML for input to the external system (i.e., requests) and for output from the external system (i.e., results).

The Database Agent connects to a relational database via Java Database Connectivity (JDBC). The SAP R/3 Agent connects to an SAP R/3 system via JCo. The Notes Agent connects to a Lotus Domino server via Domino Intenet Inter-ORB Protocol (DIIOP). The Exchange Agent connects to a Microsoft Exchange Server via Microsoft ActiveX Data Objects (ADO) and Microsoft Collaboration Data Objects (CDO). The File Agent enables access to files that are stored on file servers outside Contentbiz. It provides access to single files or to multiple files that form hierarchical folder structures. The File Agent supports http and ftp for file access.

5. Conclusion

Contentbiz today already provides a very good platform for enterprise content management and content integration.

For the future, we plan to make the following major improvements:

1) Extension of the number of agents available for other business applications.

Contentbiz is very flexible concerning the type and nature of the external business data storages and applications that can be integrated. A variety of adapters are available for different data storages and applications. Specific adapters can be built based on the Agent Server SDK. To reduce the development cost of a Contentbiz project, Contentbiz will offer a huge variety of adapters for all types of business data storage applications that will cover all market leaders in each application category. Therefore, in most projects, adapters will be available out-of-the-box, so the projects can be implemented with less cost and a shorter development time.

2) Simplified content access

Contentbiz provides several ways to access content. Content can be searched, including combined searches via meta-information stored in Contentbiz and full-text searches. Navigation through hierarchical and semantic relationships is possible. In the future, additional advanced knowledge management features such as access through categories combined with automatic classification will be added. This will be very effective, especially when there are standardized taxonomies used in a specific area. Experts on a specific topic can browse through category hierarchies that they understand. Therefore, even a user who does not know the Contentbiz application very well can easily access all the necessary information.

3) Further automization of manual processes

Contentbiz today already provides many features to automate manual processes as much as possible, for example, to automatically register new content and extract metadata. In the future, these features will be extended. For newly registered content, categories will be assigned automatically. Moreover, we are working hard on general mechanisms so that semantic information of all types, for example, relationships between

content items that describe the same customer, can be generated automatically.

4) Extended support of standard technologies

Contentbiz already supports many standards, for example, JAAS, WebDAV, and Web Services. In the near future, more standards will be supported such as Portlets (JSR168) and Content RAPI (JSR170).

These improvements will enable even more sophisticated and powerful content management and integration.

Matthias Bandorf received the M.E. degree from the Department for Computer Sciences at the Technical University of Munich, Germany in 1988. From 1989 to 1999, he worked for Softlab, where he was engaged in Repository Server Development. During this period, he designed and implemented the Transaction System of the Enabler Repository and was responsible for design and development of the storage layer of Enabler. Also, he led a joint German research project with two industry partners and two research partners in the area of component-based development. He then worked in a US-based company as a consultant supporting partners in the implementation of large CRM projects. In 2002, he joined Fujitsu Enabling Software Technology, where he has been working as Product Manager for Content Integrator.

Tsuneichi Yoshizawa received the B.S. degree in Mathematics from the University of Tokyo, Tokyo, Japan in 1985. He joined the System Engineering Group of Fujitsu Ltd., Tokyo, Japan in 1985. In 1989 he transferred to the Software Group, where he worked on the development of the Jasmine object-oriented database product. Since 2002 he has been working on the development of the Interstage Content Integrator content management product.

Yuji Takada received the B.A. and M.A. degrees from the Department of Behavioral Science and the Dr. Eng. degree from the Department of Information Engineering, Hokkaido University, Hokkaido, Japan in 1983, 1985, and 1993, respectively. He joined Fujitsu Laboratories Ltd in 1985, joined Fujitsu Limited in 1999, and since 2002 has been with Fujitsu Enabling Software Technology GmbH in Munich, Germany. In 1994 he was a visiting researcher at the Department of Computing, Imperial College, U.K. His current research interests include knowledge management, multi-agent systems, distributed computing, and machine learning.

Guenter Merbeth studied Mathematics and earned a doctorate in Computer Science from the Technical University of Dresden, Germany in 1975. Until 1979 he worked in Dresden and Karlsruhe in various research institutes and universities and in private industry. In 1979 he joined Softlab GmbH, where he worked for more than 20 years, lastly in the position of CTO for Enterprise Application Solutions. Since 2002 he has been CEO of Fujitsu Enabling Software Technology GmbH in Munich, which is a subsidiary of Fujitsu. He is currently engaged in developing and marketing products of the Fujitsu Interstage suite. His main research interests are software engineering, repository technology, and application-systems architecture. He has written a number of IT articles in professional magazines and is a co-author of two textbooks.