

Creation and Maintenance of Reuse Repository for Retrieval of Assets

T.Rajani Devi, B.Rama

Student, Department of Computer Science, Kakatiya University, Hanamkonda, Warangal, India¹

Assistant professor, Department of Computer Science, Kakatiya University, Hanamkonda, Warangal, India²

ABSTRACT: Reuse Repositories play a crucial role in an integrated reuse- based application development environment. Reusing software components implies their unrelenting storage and maintenance, and the ability to efficiently find them. Keeping in mind to built the repositories can be considered as special-purpose information systems, required to support powerful semantic modelling, flexible retrieval of varied software descriptions of multimedia nature, and efficiency optimization directed towards a large variety of classes rather than large populations per class. A repository is a necessary requirement to support software engineers and other users in the process of developing software for and with reuse. We have supposed to build- in reuse repository not only facilitate for acquiring the new assets but also should make possibly the process of quality verification.in long term ,we hope it should result in a great amount of assets with suitable high potential according to the needs of a cooperating group of people.

KEYWORDS: artifacts, repositories, software reuse, systematic reuse, quality assurer.

I. INTRODUCTION

Software artifacts are stored in a repository which supports their efficient selection. Comprehending their functionality and usage before adaptation and subsequent use in the composition of a new application is also recognized as essential in the reuse process. Adopting a software reuse repository that supports software engineers and other users in the process of developing software for and with reuse. There are some studies in the literature involving component repositories, although their focus is, for the most part, on component search and retrieval issues. Thus, different aspects, such as the roles a repository plays in a reuse process, have not been properly explored so far. In practice, repositories are used as mechanisms to store, search and retrieve artifacts, . A reuse repository once is created requires continuous maintenance. When new assets are created and have entered into the repository. The faulty assets have to be tracked. Further, when new applications emerge, the older applications become obsolete. In this case, the archaic assets might have to be removed from the repository.

Some questions raised by companies that desire to adopt a reuse repository remain unanswered. Such questions often include:

1. What are the main requirements of a reuse repository?
2. How the reuse repository must be designed?
3. What are the existing practical alternatives in it?

Repository can assume the following roles in systematic software reuse:

1. Repository as a communication bus among stakeholders: It represents a channel of asset distribution and should serve as communication bus in two ways: development for reuse (production) and development with reuse (consumption);

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

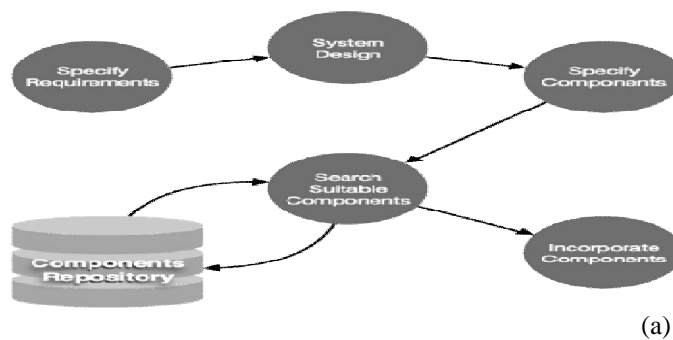
Vol. 5, Issue 4, April 2016

2. *Repository as a management assistant*: A repository can help organizations to track their assets utilization, aiming to align initiatives, projects and measure the business impact of reuse;

3. *Repository as a reuse promoter*: Repositories can propagate the reuse culture across organization by providing visibility into their assets and disseminating news about reuse initiatives;

4. *Repository as a quality assurer*: Repositories can greatly facilitate the enterprise goal of maintaining quality across the software development life cycle by providing proved components and supporting an effective component certification process.

II. REUSE REPOSITORY MAINTENANCE



Repository maintenance involves entering new items, retiring those items which are no more necessary, and modifying the search attributes of items to improve the effectiveness of search. The software industry is always trying to implement something that has not been quite done before. As patterns requirements emerge, new reusable components are identified, which may ultimately become more or less the standards. However, as technology advances, some components which are still reusable, do not fully address the current requirements. On the other hand, restricting reuse to highly mature components, sacrifices one of the creates potential reuse opportunity. Making a product available before it has been thoroughly assessed can be counterproductive. Negative experiences tend to dissolve the trust in the entire reuse framework. Reusable Assets consist of:

Packages containing elements, diagrams and structures (as drawn from any point within the Project Browser), and files in a range of text, code and graphic formats, including zip files. An asset can be ,for example:

1. A common Class library or framework
2. A set of common Requirements or Use Cases
3. A draft specification document
4. Marketing collateral

Distribution of assets in the reuse repository: The assets stored in the reuse repository are subdivided into the following categories:

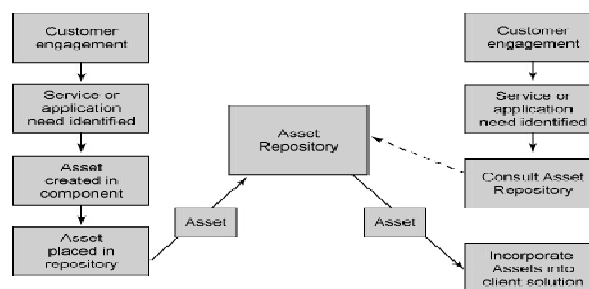
1. Internet (e.g.ASP files, HTML, pages, useful internet addresses, and so on),
2. Graphics (graphical files of different formats),
3. Multimedia (sound, video, animation files),
4. Software (files storing source or binary code),
5. Text (text files of different formats).

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

The undertaken subdivision of assets (i.e. division into some meaningful structures) allows the user or customer to match his/her needs to components obtained in the library. The category an asset belongs to makes the part of one from three classification schemes.



III. TOOLS FOR ACQUIRING THE INFORMATION STORED IN REUSE REPOSITORY

It is obvious that the effort necessary to find and recognize a component must be lower than building it from scratch. Components, which fulfill user requirements, should be retrieved as easily as possible. Browsers are important tools used to locate desired components. Unfortunately, a typical browser could not facilitate lookup through a big library. If a component is not named in a meaningful way or the library content is not organized into meaningful classes the component could be difficult to find. Using directories with precise component descriptions and classification schemes could significantly facilitate the process of identification and acquiring information about useful assets. Besides asset descriptions the directories also include references to the storage with physical assets or information how to acquire them.

In the reuse repository we have designed tools to facilitate the process of information acquiring based on active and passive strategy. By active strategy we mean looking through catalogue information through the classification schemes. The catalogue information consists of the following elements: an asset name, the category it belongs, date of introducing it into the library, short description, users' opinions, bonuses it has been granted, links to persons from several categories (author, visitor, evaluator, user, etc.).

Information acquiring based on the active strategy: It is connected with the active attitude. We have delivered special forms where a user can choose between classification schemes based on the following attributes:

1. "text" - means any sequence of characters to be matched in the assets description;
2. "period of time" - specifies the period of time when assets were introduced into reuse repository;
3. "role" - specifies a kind of person connected with assets (author, user, etc.)

Searching based on the first attribute returns information about those of assets, which have in the "description field" of catalogue the sequence of characters specified by a user. The search can be additionally qualified by a period of time. The searching returns the following information about assets: name, category, date of introducing into reuse repository resources, and asset short description. Additionally, they return links to every person associated with the concrete asset, the links to the installations, where the asset has been used, and the links making possible to copy

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

assets. The searching “from person to asset “retrieves persons fulfilling specified criteria: then a user can navigate to assets associated with these persons, according to their role in a particular asset. The system gives the link, which makes it possible to send e-mail to a selected person, for example to get more detailed information about an asset.

Information acquiring based on the passive strategy: In this mode the user waits for information. The user has the possibility of determining asset categories being in the field of his interest. A message sent from reuse repository to users is connected with some predefined events: adding new category of assets, removing a rarely “visited “category, new version of some assets, and so on. The list of events, which trigger messages, could be expanded during the development of a reuse repository is sent to users by e-mail. When building a reuse repository the set of requirements that should be consider.

IV. REPOSITORY REQUIREMENTS AND ITS DESCRIPTION

Asset Description: In this the repository must define a standard element (asset) that can represent a reusable software unit of different types. An asset model should include two parts: the asset contents (set of reusable artifacts) and the asset metadata.

Insertion: Producers need to make their assets accessible for consumption and then repositories should allow asset insertion operation.

Publish specification: In some cases users should be able to publish only the asset specification (without its implementation/content). This allows developers to register interest in implementing such assets, to be notified when new difficulty arrives at the repository.

Browsing: Each asset should be grouped in different categories and it must be possible for users to browse assets through such categories.

Search: Repositories with a large number of assets must provide search mechanisms that allow users to find assets that meet their needs. These mechanisms can be a combination of the following types: free-text search, keyword, and facet-based classification.

Report generation: The repository should provide services to generate reports that allow, among other things, to get an overview of how the repository is being used.

User Notification: Users should be able to register interest in different events with the aim of receiving notification from the repository when, for occurrence, new assets are added.

Maintenance Services: The repository system should implement administrative services that allow the maintenance of the users’ inventory and also other inventories utilized by the repository, such as, hierarchies of catalogues.

Version Management: The repository should be able to store different versions of its assets

Dependency management: Users should be able to inform dependencies between assets. These dependencies represent relationship such as “uses” or “composed-of” relations.

Feedback Services: Users should be able to provide feedback about which assets they are using. The feedback services permit the identification of well-evaluated assets.

Advertisement of Services: Repositories should allow the maintenance of reuse-related news (reuse initiatives, best assets producers, most reused components and so on).

M. Support multiple sources of assets: It should be possible to store assets that point to artifacts (asset contents) stored in different kinds of asset sources that exist in the repository deploy context.

Certification process: The repository should support a certification process which assures the quality of its available assets.

Metrics: Users should be able to define, capture and measure reuse-oriented metrics and ROI models. The metrics analysis can indicate a lot of useful things about reuse and can help managers to reduce costs and measure the business impact of reuse.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

Access Control: The system should have mechanisms to limit user access to system services and asset catalogues with the aim of defining different views among users.

Change Control: Users should be able to request and accept changes to assets. Such changes requests also include bug reports.

V. ADVANTAGE OF SOFTWARE REUSE

The architecture of the reuse repository decomposed into modules and sub-modules. At the center of the architecture is the Asset Metadata Repository. This is where all asset metadata are stored and managed. The main modules of the reuse repository are, basically: (1) Infrastructure, (2) Production, (3) Management and (4) Consumption.

For each of such modules there is a set of components that work in conjunction to provide the necessary functionality. For more information about how its internal elements are combined see Fig [1]. Each module is shortly described in following:

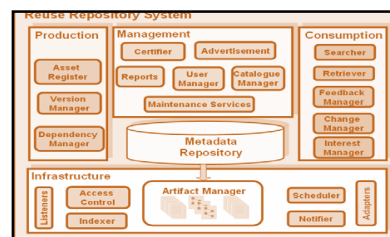


Fig [1] Repository overview architecture

1. Infrastructure module: This module groups a set of common services which are shared by the other modules. An important element of the infrastructure module is the Artifact Manager. The Artifact Manager represents a repository of asset contents (set of reusable software artifacts). Artifacts can be stored in two ways: logically - artifact content is stored in an external source; and physically - artifact is saved locally in the system.

Once the asset content and its metadata has been updated or inserted, listeners automatically start the indexing phase. During this phase, performed by the Indexer component, the artifacts and metadata of the asset are read from the Metadata Repository and the Artifact Manager, and then such data is parsed, being actually indexed. The resultant index is used to search the assets in repository. Additionally, the infrastructure module has other services grouped into the following submodules: Indexer, Security, and Notification.

2. Production module: This module is responsible for services commonly used by assets producers. Such services are provided by the Asset Register component. The Asset Registry is responsible for the insertion of assets and it interacts with the Metadata Repository and Artifact Manager components. Beyond that, the production module has two other elements: Version and Dependency Manager;

3. Management module: This module is responsible for all services used by repository administrators and asset/component certifiers. The management module is composed of six other submodules: Certification, Advertisement, Reports, Analytics, Access control, and Maintenance services.

4. Consumption module: This module is responsible for services commonly used by assets consumers. Such services are grouped into the following submodules: Search/Browsing, Change manager, Feedback Retrieval.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

VI. DEVELOPMENT PROCESS

The software development process must be enhanced to include reuse activities. A reuse library or repository must be created and maintained, and tools must be acquired or developed to access the assets, and many new procedures must be specified:

1. Procedures for developing reusable assets and inclusion of assets in the repository
2. Procedures for domain analysis and architecture design and modification
3. Procedures for configuration management and control of reusable assets

Project planning should include extra time for designing, implementing, and testing robust reusable assets as opposed to system-specific functionality, since their quality is important not just to one system, but potentially many future systems. Time must be allotted to researching repository assets to be included for reuse and matching them to requirements. The key activities are the following:

Software product and/or process requirements are evaluated to determine if existing software assets exist that can fulfill the requirements. (i.e., matching needs to capabilities) .

What is an Asset ?

An asset can be defined as anything of material value or usefulness. There are several types of assets:

Best practices: Techniques or methodologies that, through experience and research, have proven to reliably lead to a desired result.

Designs: Written documents that describe a general solution to a design problem that recurs repeatedly in many projects.

Tools: Set of assets that aid in development or recreation of a project, application or task.

Components: Identifiable parts of a larger program or construction.

VII. REUSE PROCESS MODELS

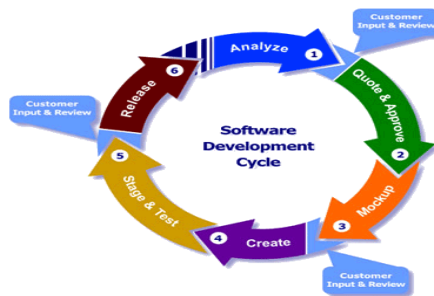
Assets are identified and evaluated for reuse.

1. Asset certification requirements are established to determine asset completeness, quality, and/or history.
2. A library (ies)/repository (ies) of reusable software assets is established and maintained.
3. The software reuse activities are maintained, managed, and controlled as part of the organization's and project's defined software process.
4. Incorporation and/or development of reusable assets are included in the project's software costing and sizing practices.

International Journal of Innovative Research in Science, Engineering and Technology

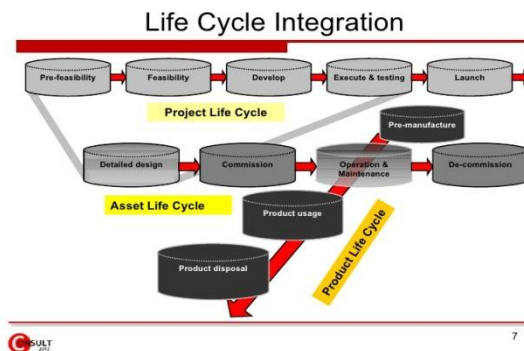
(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016



reuse life cycle

Reuse must be considered through all phases of project life-cycle. Partial adoption of reuse strategies is not enough. Opportunistic reuse does not allow for the organization-wide standardization and control necessary for the maintenance of a true core repository



VIII. CONCLUSION

Repositories of software assets must be created and maintained. Reusable assets must be designed for reusability. In assumptions, the reuse repository allows gathering of reuse assets and accessing them by a collaborating group of users or customers. Every group member has rights to decide on reuse resources, voting after or against. In that sense, every user can be treated as a repository co-author. Reuse repositories can be useful for every group of people using computers and cooperating through network. Depend on specialization (or interests) of a group as a whole or of its individual members the reuse repository can include assets belonging to different categories. Exemplary assets, for a software developers' team, could be function or class libraries, subsystem or application frameworks, design patterns, and so on. Furthermore, for a desktop publishing group a possible set of useful assets could include: graphical files, advertisement slogans, document patterns, etc. During designing of the system we haven't put any limitations on assets categories. We have also provide in the system special tools for managing assets categories which can support the information of the reuse culture. A reuse repository once is created requires continuous maintenance. When New assets are created and have entered into the repository. The faulty assets have to be tracked. Further, when new applications emerge, the older applications become obsolete. In this case, the obsolete assets might have to be removed from the repository.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Vol. 5, Issue 4, April 2016

REFERENCES

- [1] [LG99] Luqi and Jiang Guo, Toward Automated Retrieval for a Software Component Repository Proceedings of IEEE Conference and Workshop on Engineering of Computer-Based Systems, March 1- 12, 1999, Nushville, Tennessee, USA.
- [2] V. Burégio, "Specification, Design and Implementation of a Reuse Repository", MSc dissertation, Federal University of Pernambuco, Brazil, 2006.
- [3] M. Morisio, M. Ezran, and C. Tully, "Success and Failure Factors in Software Reuse," IEEE Trans. Software Eng., vol. 28, no. 4, pp. 340-357, 2002.
- [4] Burton, Bruce A. et. al., "The Reusable Software Library," IEEE Software., July 1987, pp. 25-33.
- [5] Y.S.1 Ma arek, D.M. Berry, G.E.Kaiser: An Information Retrieval Approach for Automatically Constructing Software Libraries, IEEE Trans. on Software Engineering, Vol. 17, No 8, (August 1991).
[CON93] P. Constantopoulos, M.Doerr and Y. Vassiliou: Repositories for software reuse: the software information base, N. Prakash, C. Rolland, B. Pemici,eds. Information System Development Process, pp. 285-311, (North-Holland, 1993).
- [6] S. Henninger. An evolutionary approach to constructing effective software reuse repositories. ACM Trans. Softw.Eng. Methodol., 6(2):111–140, 1997.