A Detailed Study on Application of Clear Case in Versioning Software

Agha Salman Haider

Research Scholar (Computer Science), CMJ University, Shillong, Meghalaya

Abstract— For a software configuration management system to support large scale development projects, it must address the difficult problem of geographically development. In this regard software utility tools like clear case in eradicate such problems to a considerable extent of sorts which is a complete software constitution management system. It manages various variants of evolving software systems, track whose various forms were incorporated to form software builds. In a large development organization the software developers are located at different physical sites and each site may develop sub components of a larger system. In this regard sites need to share resources and tools like clear case can help things to a considerable extent.

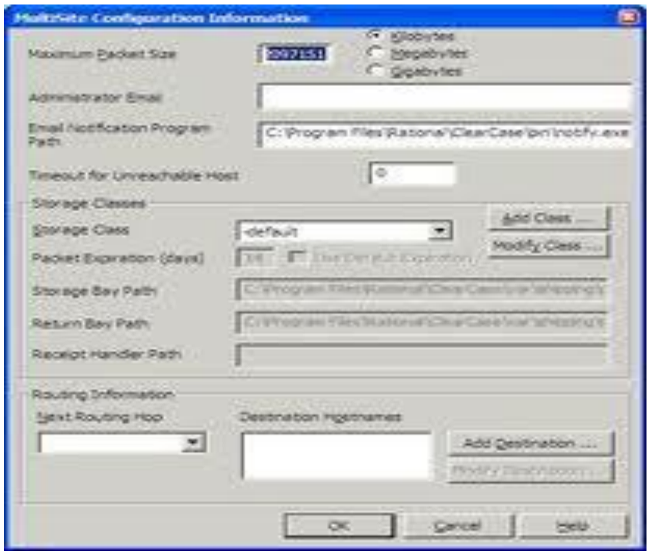
---♦-------------------------------------

1. INTRODUCTION

The size as well as the complexity of the software projects have increased over time. It is quiet common to find a single software system with millions of codes under development by several hundred software engineers. The clear case product provides configuration management and software process control for parallel development in a local area network. The main utility of clear case is to provide geographic distributed parallel development. In this regard there is a need to understand the concepts of clear case first and then one can proceed to the multiple variants of it.

After 20 years of its launch, Clear Case is one of the major software configuration tools in the market. In fact the concept of SCM is not clearly defined as things ought to have been in the first place. Clear case is still an original as well as a valuable product in the world of today. It can be used and shared by anybody and not reserved for individual specialists as things stand (Girod and Shpichko, 2011). The second birth of configuration management is known as version control or known as source code management which took place in the early days of the software industry that is the period around 1960. The third period of birth is known as SCM which occurred in the decade of 1970's. In a way it could be referred to as the merger of the two divisions earlier. Things become simpler as well as easy to understand with the development of tools like UNIX and C. In fact two major ideas strike clear case as an SCM tool which is

- Presenting the version database via a file system interface
- Audit building transactions and recording the system calls for opening file objects which leads to a graph of dependencies being produced.



Maximum Packet Size: 2097151 [✓ Kilobytes]
Administrator Email:
Email Notification Program Path: C:\Program Files\Rational\ClearCase\bin\notify.exe
Timeout for Unreachable Host: 0
Storage Classes
Storage Class: -default
Packet Expiration (days): [illegible]
Storage Bay Path: [illegible]
Return Bay Path: [illegible]
Receipt Handler Path:
Routing Information
Next Routing Hop:
Destination Hostnames:

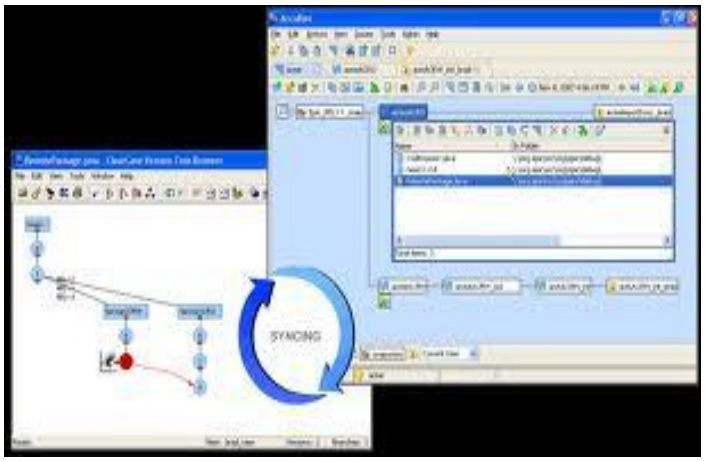
CASE tools can be conferred to as engineering tools to help in the aid of software development. It happens to be tool for the end users. It can run on any sort of platforms and can store large files of any size. In addition to this it supports branching as well as labelling. Atria software is credited with the development of clear case. In traditional terms clear case supports full as well as fat clients. From the point of view of developer if any file is created in clear case then it will refer to the new directories in question and it allows one to build systems with the same code in operation as well.

THE MEANING OF THE TERM CLEAR CASE IN VERSIONING SOFTWARE

Many programmes as well as projects use different names for clear case roles as well as assignments but configuration management can use the following as an example

- Architect- understands and sets up the system architecture
- Configuration management- familiar with the change as well as the control process
- Lead- responsible for schedule and changes
- Software design engineer- makes changes to file under the configuration system
- Build engineer- utilizes software build concepts as well as tools.

In fact a functional overview of a clear case concept is the repository named virtual object base (VOB). All files as well as directories are managed inside the VOB and can expand from hundreds of files as well as the directories to thousands. On the other hand the files and directories are transferred to other repositories when the database becomes too large. In other words they can also be split and made to work together as well. This ensures the check out of files and supports data recovery when it is needed (Summers, 2012).



The SCM tool provides support for change management request and is a complimentary tool for clear case. The database utility is used for recording, tracking as well as reporting about the internal mechanisms. The change request is from the team members to change a process. Documented in this type of request is the information of problems which can occur during software development along with the impacts which could occur. In this regard all the changes of software are documented and approved as the terms stated may be.

The clear case tools provide an open architecture to implement configuration management as well as control solutions. The web site content for military as well as aerospace industries employs many different development environments (Summers, 2012).

As long as the test changes are being made to the software it cannot run properly. Clear case is a complex system and it takes a lot of time to understand the advantages over simpler systems in our setting. This is not obvious at the beginning stages of the project, but is clear as the project evolves over time. The cost of the system has proven to be quiet insignificant with the potential positives which evolve out of it. In fact due to the cost factor involved clear case may be suited for large companies which support a variety of projects on numerous platforms. On the other side of the coin clear case is definitely a good investment option in case of companies which are medium sized and have 8 to 12 people at their disposal. This is both from the business terms as well as improving the work as well as the environment.

If one uses a CASE tool to develop the design, much of the work will be performed automatically by the CASE software. There are many ways of proceeding with the task model and experience will be the best teacher. If one does not prefer to use a CASE generated model, then each diagram as well as object definition should be supported by clear and relevant documentation that can be easily analyzed by any person who reviews the model. In this regard the diagrams as well as the documents are the foundation for the system’s design, so accuracy is all the more important. It needs to be kept in mind that is much easier to repair a diagram then change software later. So the message which comes across from the tool of clear case is that one needs to anticipate changes at each and every point.

ADVANTAGES OF CLEAR CASE OF VERSIONING SOFTWARE

When clear case was launched in 1992, it made quiet a serious number of changes on how versioning control was used. Unlike the previous systems clear case integrates into your operating system and this contributes to make the version experience seamless. In the year 2003, IBM brought clear case and they still have it. There is a lot of complexity involved in setting up of clear case as far as configuration spaces or setting up internal network of spaces. For the clear case probably you need to be a large company with a couple of network engineers at your disposal to take advantage of it. Clear case builds automated software builds, an automated merging system, an extensive network, distributed leads and a whole lot of things. Of course all this is not that cheap and the cost of setting up clear case is not that cheap also. If one is planning to set up they need to consider at least a few months at their disposal. However despite all this Clear case is one of the market leaders and is used by various corporations all over the world. In fact all the negatives are worth considering if one has sufficient money as well as time at their disposal (Kemper and Oxley, 2012).

IBM Rational ClearCase Remote Client V7.0.1 Uninstaller

Welcome to the InstallShield Wizard for IBM Rational ClearCase Remote Client V7.0.1

The InstallShield Wizard will uninstall IBM Rational ClearCase Remote Client V7.0.1 from your computer.
To continue, choose Next.
IBM Rational ClearCase Remote Client V7.0.1

Versioning is the most important capability which an SCM system provides. The SCM system stores a complete revision history of every change made to the file. Each modification is assigned a specific way and you can access them in a convenient way you want. The whole idea is if you see a change and then discover some unintended consequences, it is quiet revert to the oldest version of the working systems. In most SCM systems you can also use tags to mark specific versions of a code. This is all the more important if one is planning the release of their software. One can roll back to the tagged version just as they can do to a revision number. The SCM helps one to collaborate and coordinate in a team environment. Whenever a developer makes a change, the system stores metadata about you made the revision and when it occurred. It provides a shared space that helps the team document and share designs and ideas. Documentation is important when the things need to be kept up to data and when it is near to the code. If the team focuses on the documentation it is more likely that the things will be up to date on all counts. Other than this it will have the same revision history at the code. This means that if one has a particular date or tag, they will get proper versions of both the code as well as the corresponding documentation (Hibbs, Jewett and Sullivan, 2009).

Many case study tools provide limited functionality to a user. Many case tools are single end tools that mainly aid in the construction of an aid or a drawing. This is generally an improvement over the manual method of drawings. In this regard CASE tools are attempting to grow and provide high end reliability such as access control development, development and support for multiple concurrent users as well. While most operating systems are the most relevant system of system software, the great majority of software which is developed is sold for individual profit and it is used for one's own purposes as well as business purposes (Oz, 2006).

APPLICATION OF CLEAR CASE IN VERSIONING SOFTWARE

Clear case is among the most popular as well as sophisticated version control mechanisms. It has conceptually powerful development as well as branching capabilities. As already stressed upon Clear case has been developed by rational software and now it is owned by IBM and it provides life cycle management as well as control of software assets. As far as the icing on the cake is concerned with an integrated system and parallel development support it provides opportunities to retrieve as well as update the files (Jielin, n.d.).

The use case analysis helps in understanding the problem space better. It gathers with the problem definition and starts with collecting of information which is based on the behavioural traits. In fact the major objective of case analysis is to document the business process that need to be ultimately supported by the system under the development. They are developed in mind considering the following goals as well

- It should enable communication among all the stake holders- This process begins when the users communicate their needs to the developers and help the customer verify the fact whether their requirements are met or not. Through this medium the software developers communicate how the business requirements will be met by the system. In this regard the concept of mapping comes into the picture which matches specific functions to the cases. In the cases where mapping is not possible the solution does not meet the requirements. This is also all the more handy when there is a disagreement between the customer and the developer after the completion of the software solution (Varma, 2009).

- User cases must justify business process as well as actions- this will clearly define the objectives of the business which needs to be achieved by the system. The business objectives are achieved in the following manner. In the first place all the processes are documented and one needs to understand each and every step of it. Using this approach one can eliminate all the unnecessary steps and work towards achievement of the goal of the organization. Secondly each and every step whether it is automated or manual should be clearly known as it helps in better interactions as well as communications.

Clear case helps the organizations to manage their change process better. It provides version control, workspace management, and parallel development support and build audit to improve productivity (Sage and Rouse, 2009). Some of the major applications of clear case solutions are as follows

- It can allow one to work on the global as well as the local platform
- The model is based on proven text practices which have contributed to the efficiency as well as productivity
- It can be operated in diverse platforms and the cross platform mechanisms can extend across various teams
- Operating system work on various platforms like UNIX etc

FUTURE OF CLEAR CASE OF VERSIONING SOFTWARE

CASE as the name signifies stands for customer aided systems engineering. These are case tools that software developers use to help them design and construct information systems. The CASE tools can speed up development, reduce costs as well as provide an inclusive documentation which can be used for future maintenance as well as enhancements. In fact the older systems had to rely on programme code which used procedural language such as COBOL which required a programmer to create code statements for each and every processing step. On the other side of the coin modern languages such as C and Java are non procedural where a programmer must perform when certain events occur. In this regard an integrated set of CASE models can be used to model, document, engineer as well as construct the information system. A CASE tool can handle many programme development tasks such as generating application codes, screens as well as reports. To consider it from another point of view an integrated development environment (IDE) has an in built CASE tool that a software vendor includes to make it plan, construct as well as maintain a specific product service. Some of the examples of IDE's are Oracle Designer as well as the Microsoft Visual Studio 2008. Two things are clear from all the analysis as the CASE tool vendors will continue to include powerful new features and the popularity of such object oriented tools will continue to grow all the more in the days to come (Shelly, Cashmand and Rosenblatt, 2010).

Being able to produce a build is great and to repeat it is also an option worth considering. Repeatability is the same thing being produced over and over again and reliability is of being correct at the same time. Lack of both these factors can be manifest in a number of ways (Lee, 2006).

The information should be placed into a file so that it can have a repository for consultation as well as analysis of future change requests. In this regard a configuration tool like Clear Case can make the required number of changes which may be beneficial in the long run. So this brings us to the fundamental question on what is the utility of a configuration tool. It allows the tracking as well as managing all pieces of software artefacts. In a way it can also be applied to the process of changing of control request of all forms (Tsui, 2011).

Even experienced engineers are surprised by the fact on how much effort is required to implement a simple change. The complications may not arise from the first stages of the project but may pave way for a ripple effect- the cost and effect relationships on schedules as well as costs. Software development teams have used software configuration systems (SCM) for many users and the main utility of this system is that it allows multi users to work on the same software while reducing the chances that the individual will overwrite each other's work. It provides a mechanism for identifying the files that make a release also. The tool of clear case allows the developers to monitor the versions of the file, share files during the development stage and label a file for a set of release. In addition to this tool also contains the history of changes that have been made to the file also. The key point of consideration in this regard is to understand how the SCM history can be tapped to gain an insight into the activities of the software team along with the software product itself. All this provides for a rosy future in the long run (Caroll and Daughtrey, 2007).

CONCLUSION

From the analysis of the tool clear case and its application in the domain of version software a few facts emerge. All said and done the basic fact is that the cost involved is on the higher side, but if implemented is bound to yield results and supersedes the cost factor. There is lot of misconception to the fact that is ideal for only large companies with big platforms but the reality is far from truth as small and medium sized companies can also implement it and this contributed to the improvement of the business and the environment in which it operates.

Other than this the cost of introducing as well as implementing clear case was recovered in less than a year and this was much less than the time frame expected. In addition to this introducing clear case was not as easy as expected and in spite of careful planning and experimentation it caused more problems for the developers than was expected. There has to be a well established system for handling errors which contributed to the success of the project big time. In the final analysis the software tool for change management will evolve more and more in the days to come.

REFERENCES

1. Girod, M., and Shpichko, T. (2011). IBM Rational Clearcase 7.0: Master the Tools That Monitor, Analyze, and Manage Software Configurations Lincoln Road: Packt Publishing.

2. Summers, B. (2012). Effective Methods for Software and Systems Integration. Boca Raton: Taylor & Francis

3. Kemper, C., and Oxley, I. (n.d.). Foundation Version Control for Web Developers. Apress.

4. Hibbs, C., Jewett, S., and Sullivan, M. (2009). The Art of Lean Software Development: A Practical and Incremental Approach. CA: O'Reilley.

5. Oz, E. (2006). Administración de los sitemas de información. Canada: Thomson

6. Javvin, and Jielin, D. (2012). Network Dictionary. Javvin Technologies

7. Varma, V. (2009). Software Architecture: A Case Based Approach. New Delhi: Dorling Kindersley

8. Sage, A., and Rouse, W. (2009). Handbook of Systems Engineering and Management. NJ: John Wiley

9. Shelly, G., Cashman, T., and Rosenblatt, H. (2010). Systems Analysis and Design. Cape Town: Cengage.

10. Lee, K. (2006). ClearCase, Ant, and CruiseControl: The Java Developer's Guide to Accelerating and Automating the Build Process. Adobe Press.

11. Tsui, F. (2011). Managing Systems and It Projects. MA: John Bartlett.

12. Carroll, S., and Daughtrey, T. (2007). Fundamental Concepts for the Software Quality Engineer, Volume 2, Volume 2. Milwaukee: American Society for Quality.