

## GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

### IMPACT OF COTS COMPONENTS IN SOFTWARE UTILITY

N. Gnanasankaran <sup>\*1</sup> & V.A. Anand <sup>2</sup>

<sup>1</sup>Dept of Computer Science,

<sup>2</sup>Dept of Logistics Management,

<sup>1&2</sup>Alagappa University, Karaikudi, Tamilnadu – India

#### ABSTRACT

Commercial Off-The-Shelf (COTS) technology is widely used in many industries and also in scientific computing. In this paper, an introduction to the usage of COTS components and its definition are given. The virtues of COTS and the skills needed for using them are furnished. The importance of integration and composition of COTS are touched upon. Finally, the drawbacks are indicated. Conclusions including the future scope of COTS are also given.

*Keywords: COTS, Software Integration & Composition.*

#### I. INTRODUCTION

The cost of software development can be significantly reduced by fostering a software component industry. In the past, most software were marketed as complete applications; today we are moving toward an economy where it is hoped that components can perform dual purposes and thus be sold as stand-alone entities. When components can be marketed in this manner, they are referred to as “Commercial Off-The-Shelf”(COTS). It’s interesting to note that more than 99% of computer instructions come from COTS products (Basili, 2001). It’s inevitable that your SW for any activity will involve COTS software. Your project team simply cannot write every component needed to make an effective SW-related decisions and analyses. But, your evaluation of COTS is important so as to minimize risks and maximize benefits within the scope of your projects.

Almost without exception, every software-related endeavor will utilize a significant percentage of COTS software components. In this article, the definition of COTS, its virtues, skills needed to use it, integration of COTS components and the drawbacks of COTS SW are detailed.

#### II. COTS – DEFINITION

- A COTS product is sold, leased, or licensed to the general public and offered by a vendor trying to profit from it;
- supported and evolved by the vendor, who retains the intellectual property rights; available in multiple, identical copies and used without source code modification

#### III. USING EXISTING SOFTWARE, INCLUDING COMMERCIAL OFF-THE SHELF SOFTWARE

In the 1970s and 1980s, software reuse was identified as a key strategy for reducing software costs. This was known as “Japanese software factory” approach- Design (architecting) for reuse.

In the 1990s object-oriented software makes integration of separately developed software components more feasible. The software profession has got reusable components, but they don’t meet industry criteria for appropriateness for reuse. Design for reuse “the right way” seems too expensive for commercial software developers, just as it was for system developers.

**IV. VIRTUES OF COMMERCIAL OFF-THE-SHELF**

- Costs are lower than custom development, because product development costs are shared over many users
- Many others participate in finding bugs and limitations to the product (and the producer may actually fix these bugs)
- You can incorporate new technology more quickly because you use products containing it without having to learn all about it yourself
- Development time and risk are avoided when the COTS product provides all the features you need.

**V. KEY SKILLS IN USING COTS SOFTWARE**

- Different skills are needed for using COTS Software effectively. Most COTS SW packages could be designed/implemented better by your good people [1].
- Designing and writing code are more fun for most software engineers than integrating the (suboptimal) packages available off-the-shelf
- Integration skills are different than development skills
- Problem-resolution is different when you must work around limitations/characteristics of a COTS SW package.

**VI. DRAWBACKS OF COTS SOFTWARE**

- Harmonization of a COTS package with
  - Platform-specific operating system variants and peripheral drivers
  - Operating system version(s)
  - Companion COTS SW packages
- Feature availability and timing (vaporware)
- Bug fixes often only available in later releases
- Difficult features dropped in later releases
- Features glut can swell resource requirements
- Long-term support of COTS software
  - Company survival (escrowing source code)
  - Evolution of features.

**VII. INTEGRATION/COMPOSITION OF COTS COMPONENTS**

When we look at developing software systems using components, it is the work of integrating the components with each other and the rest of the system that is the most important part of the component-based development process. Depending on the component model used and the actual components that are to be integrated, more or less work will have to be done in order to get all the parts of the system to function correctly together. It is this composition of different parts that make reasoning about quality attributes in a component-based system more complex than in a less modular system. Each component will have its own quality attribute profile, but when interfaced and used together with other components, the resulting composition may show a different quality attribute profile altogether.[2]

When integrating components into a system assembly, it would be useful to be able to predict how the quality attributes for the whole system will be. The predictability of a composed system depends on many variables, not just the components or the component framework. The quality of the extra code needed to integrate the components into a functioning system (the glueware) will also affect the system, and perhaps even more importantly, process and organizational factors will also affect the system.[3]

## VIII. CONCLUSION

- In the future, most software development efforts won't be able to afford not to use off- the-shelf software
- Used inappropriately, off-the-shelf software can cost more to use than developing needed software functionality from scratch
- Understanding what it takes to use COTS SW effectively is very important, so you can help your enterprise make the right business decisions.[4]

## REFERENCES

1. N. Gnanasankaran, S. Natarajan, K. Alagarsamy and K. Iyakutti, *Application of COTS Components: A Software Package for Crystallography*, *British Journal of Mathematics & Computer Science*, 3(2):99-107,2013..
2. N. Gnanasankaran, S.Natarajan, K. Alagarsamy and K. Iyakutti, *A case study of the application of COTS components in a molecular dynamics software*, *Paper presented at the Fourth International Conference on Electronics Computer Technology (ICECT 2012)*, Kanyakumari, India, 6-8 April, 2012.
3. N. Gnanasankaran, S. Natarajan, K. Alagarsamy and K. Iyakutti, *Commercial Off-The-Shelf(COTS) Components in Software Engineering: The Software Package SCILAB*, *Int. J. Computer Technology & Applications*, 4(1), 68-71, 2013.
4. L.Brownsword,T.Oberndorf,C.Sledge.*Developing New Processes for COTS Based Systems*, *IEEE Software*, July/August 2000, pp. 48-55.