

# Evolution of Component Based Systems

*Pearl Brereton  
Department of Computer Science  
Keele University  
Keele  
Staffordshire  
ST5 5BG  
UK*

Email [o.p.brereton@cs.keele.ac.uk](mailto:o.p.brereton@cs.keele.ac.uk)

March 1999

## **1. Introduction**

Many organisations are moving towards a component based approach to software development. However, there is a significant risk that component based systems will become the legacy software of the future. The difficulties of maintaining systems for which responsibility is distributed across many authors, owners and organisations is aptly illustrated by the increasing World Wide Web (WWW) maintenance mountain [1].

The strengths and opportunities associated with component based development stem from the potential for reduced costs and increased functionality and quality that multiple suppliers (of components) can bring. Potential benefits also accrue from reuse. These are well documented although considered by many to be slow to materialise. In addition, diversity in the solution space should improve component integrators' options to trade requirements against cost, delivery time and/or other factors such as component quality and supplier reputation.

The commercialisation of software components can be expected to widen the range of drivers for software change. Traditionally, software change has been driven externally by customer and market requirements and internally, by the need for corrective and adaptive development and maintenance. For component based systems (CBS), we might add to these, the push by vendors to stimulate change by offering new improved (or cheaper) components or by withdrawing support for components already in use. Similarly, integrators may choose to move to preferred suppliers or away from risky or blacklisted suppliers. In this way, the supply chain becomes an important dependency to be accommodated by change management systems.

The evolution of CBS, like that of WWW documents, needs to be strictly controlled if such systems are to maintain their initial levels of quality throughout their operational life.

This position paper aims to identify the major CBS maintenance issues and to suggest areas of research needed in order to address these issues. It aims to contribute to sections two and four of the Strawman outline.

## **2. Maintenance Issues**

CBS maintenance issues are listed under the headings: *business*, *management* and *technical*.

### **Business issues**

**Responsibility for change** - The nature of software makes it notoriously difficult to separate out the source of a particular fault even when an in-house team produces the elements. For CBS it will be important to establish sound methods of assigning and enforcing responsibility for parts and for the 'whole' system. Integrators (and customers) of CBS may chose to work with a limited set of 'preferred suppliers' rather than acquiring components on the open market.

**Risks of change** - The many risks traditionally associated with change are likely to remain, or even to increase, for CBS. For example, analysing the impact of replacing components by others from different suppliers will be a more complex task than undertaking impact analysis 'in house'. Risk analysis is not a widespread software engineering skill.

**Payment for change** - A number of issues relating to payment and charging are likely to arise and it is possible that billing will become a major overhead for the component based software industry. If, for example, customers pay for components on a per use basis (where such use may involve remote execution), payment models may have to incorporate payment to component providers as well as to systems integrators. A much broader range of payment models than are used at present may be needed in the future to accommodate both the complex webs of owners and agents as well as different purchasing and licensing models.

**Future proofing** - The potential for providing long-term support is likely to be a major factor to be considered when purchasing components (since history suggests that even 'throw-away' systems can remain in use for quite long spans of time!). Employing a mechanism such as escrow (keeping source code with an independent, trusted and secure repository) may help with customer reassurance. The use of such practices as employing multiple sources (as with hardware systems) and using preferred suppliers may provide further reassurance.

### **Management issues**

**Drivers for change** - CBS maintainers are likely to be subject to disparate and potentially overwhelming demands for change. On the one hand, vendors of components will continue to produce and market new improved components and may also either withdraw support for components in use or adjust charging such that change is unavoidable. On the other hand, customers, as today, will always require new features or facilities. In addition, vendors of integrated systems are likely to strive to identify new markets in order to extend their portfolio of products and services.

All three of these 'drivers for change' imply the need for quite different practices to those employed to produce 'bespoke' systems (tailor-made for the individual customer) or packaged systems such as word processors and spreadsheets.

**Change policies for distributed systems** - The assumption in this paper so far has been that components of a CBS are physically integrated to provide an executable system to run on a customer's computer. However, current technology supports the development of virtual (integrated) systems where components remain at the provider sites (or at some other remote site) and are accessed as required. Such systems could reduce some of the problems associated with system upgrades and version management. Upgrading a component of such a system could be carried out by 'simply' replacing it by another. Possible upgrade policies (or strategies) might include:

- Replace component with a new version (no notification to users)
- Replace component with a new version and notify users
- Give users a choice to continue using the 'old' component or move to the new one
- Require new users to use only the latest version of a component

**Component documentation and description** - information about components will need to be accumulated and combined. Such information will come from a number of sources and will have a range of forms (e.g. factual, opinion, statistical). Sources of information might include:

- component suppliers - e.g. advertising literature, component introspection
- standards bodies and certification organisations - e.g. certification of compliance to interface standards
- special interest groups - e.g. component reliability measures from other users
- integrator organisations - e.g. assessments through benchmarking, case studies, experiments

## **Technical issues**

For CBS, the fundamental unit of change is at the component level. Change includes:

- upgrading a component to a new version, provided by the same supplier
- replacing a component with one from a different supplier
- adding a new component
- removing a component

Maintenance activities will include:

- *locating, understanding* and *evaluating* potential replacement or additional components
- *determining the impact* (on the overall system) of potential change
- *estimating the cost of re-testing*
- *evaluating risks* associated with using new suppliers

## **3. Key research areas**

>From the issues discussed, the following CBS maintenance research topics emerge:

- **Evaluation** - including evaluation of components, products suppliers, development and maintenance strategies and alternatives, architectures, risk, productivity, skills.
- **Interaction and integration of business and technical factors** - relating to, for example, selection of components and suppliers, cost/functionality/quality/availability/confidence trade-offs, future proofing, managing change, payment models, integration of business and technical processes.
- **Component descriptions and documentation** - including the range of description forms, sources of descriptive information, maintenance of descriptions, users and usage of descriptions and description quality.

## References

1. OP Brereton, D Budgen and G Hamilton, *Hypertext: the next maintenance mountain*, IEEE Computer Vol. 31, No. 12, December 1998, pp 49-55