

# Project Development and Componentisation - an Introduction and Discussion

*This article explains that it is important to use quality suppliers and that the use of embedded software components provides the most cost effective and time effective way of developing embedded projects. Using embedded software components also provides significant competitive advantages in terms of ensuring quality and advanced features.*

## BASIC CONSIDERATIONS

In developing a project the Project Manager has three basic considerations - quality (not only adherence to project requirements but an overall build quality meeting future considerations and requirements), speed of development (productivity) and cost of development. These three issues are directly related to the personnel used for the project development and the type of company in which the work is undertaken.

One of the most important factors that influences quality, productivity and cost is motivation, or more correctly a clear answer to the "Why bother?" question. Within companies one typically has a combination of permanent and contract staff. These are two very different types of people. The permanent staff member doesn't have any of the business worries and large expenses of the contractor but he, or she, does have stability of income, even if this is at a slightly lower level. The permanent staff member also has a good knowledge of the company's area of business and technology. The contractor, on the other hand, has a broad range of experience and is very familiar with quality issues, if supported and managed correctly is highly productive and is consequentially very cost effective. The contractor, however, takes a little time to get up to speed - the learning curve. Contractors tend to be much more highly motivated than permanent staff. They have often taken on contracting in answer to the saying - "Stay too long with a company and you become part of the furniture and get sat on".

The problem with contracting, though, is that the contractor is often working on someone else's "dream". This is where the component supplier comes in - a contractor working on his, or her, own projects. There is nothing more motivating than working on one's own projects. Consequentially, within a component supplier, quality and productivity are maximised and cost is minimised.

So the obvious answer to these issues is to use a small group of incentivised, and consequentially well rewarded, permanent staff, in a good company environment and good management structure, to handle the issues of project management - writing of require-

ment specifications, design of software and hardware architectures, sourcing of software and hardware components, integration and configuration of the software components, construction of the hardware and project completion and testing.

## OTHER FACTORS

The component supplier has already constructed the components so has removed all the unknowns from the project development. The code is already developed and tested and as such is proven. It is mature code.

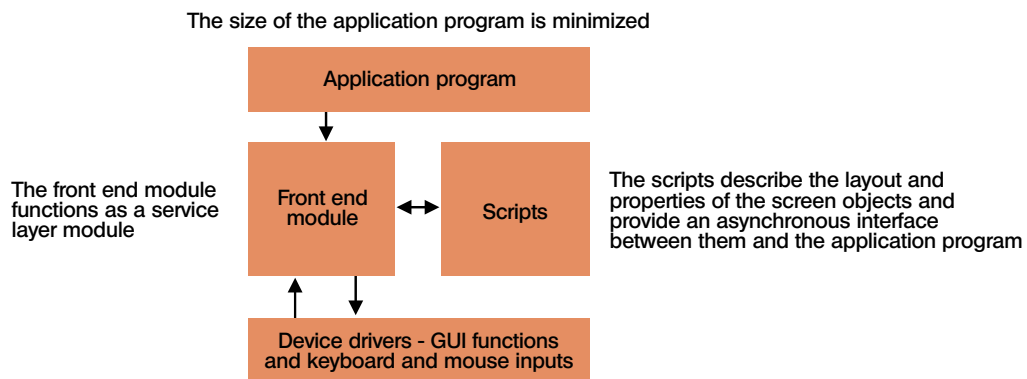
Development times using the components are clearly known and if the components are of quality they are very minimal. Further the code does not need to be redeveloped - the wheel does not need to be reinvented. As such

project development time is greatly reduced. The components just need to be set up and configured. Because the unknowns in the development time have been eliminated - all the hard work has been done - the development time and costs can be closely managed.

Connected work - upper and lower layers etc. - can proceed much sooner thus further reducing the development time of the project.

Because both the development time and cost are greatly reduced the competitiveness and return on investment of the company is maximised. The competitiveness is further increased when quality components are used.

The use of components means that the project is designed with a definite architecture. As such the interfaces are clearly designed and the inter-component relationships are clearly defined. Componentising a project means that all problems are localised, identified and solved. The project is designed in a manner that is easily re-configurable and expandable. This means that the project is easy to evolve, improve and upgrade. A change in one component usually does not effect other parts of the program. Extra features are easily added by just substituting the existing components with new components. Components have to be general. As such they have a far greater life time than



The front end module provides a simple and quick way of designing highly effective user interfaces

Figure 1.

project specific code. This means that ongoing costs and development times are greatly reduced.

A high level of knowledge and expertise is brought directly into the project and the company. The component supplier has experience over a large number of projects and clients and brings this experience directly into the component construction and supply. New knowledge, expertise and features are brought in as they become available. Companies starting off often don't have the time and resources to gain a large technological edge over their competitors. The usage of components allows them to gain this advantage.

The link with the developer can be maintained thus ensuring that code maintenance and updates are easily handled. If external contract staff are used for code development there is no guarantee that they will be available subsequent to the project completion to maintain and update the code. If a component supplier is used maintenance and updates are supplied as part of the support service. Support and maintenance costs are minimised - there is only one piece of code to maintain.

Quality is guaranteed. Further the specification of the usage of specific component suppliers by the customer ensures that the project is designed to the desired level of quality and capability. Component suppliers are usually highly motivated. It's this motivation that ensures that you get the best product. Component suppliers are often prepared to expand the features and capabilities of their products in order to account for new market areas.

Because the code is componentised the code is easily audited. Issues such as safety and security are easily handled.

Copyright issues are not effected. Software component copyright can be viewed in the same way that hardware component copyright is handled - the company owns the overall design copyright - the circuit diagram - the chip suppliers own the copyrights of the individual chips - the hardware equivalent of the software components. The components can be used without effecting the saleability of the company's products whilst maintaining the original copyright of the component supplier.

## CURRENTLY AVAILABLE SOFTWARE COMPONENTS

The following is a list of the major areas in which software components are currently supplied in the embedded software market:-

### FRONT ENDS

Also known as Man Machine Interface's, Human Interface's and User Interface's. These provide a simple method of linking the user to the application program. They fall into two categories - Binary Coded Script based and Text Coded Script based (eg. HTML). The Binary Coded Script based Front Ends are the most compact both in terms of the module and the scripts. They also operate faster, link to the application easier and provide more control over the layout and presentation. Front Ends are supplied by U.K., Continental European and U.S. companies.

### ARCHITECTURAL RELATIONSHIP

The Front End Module provides a simple and quick way of designing highly effective user interfaces. (See Figure 1)

A Front End Module also allows the separation of the application into a functional module and a presentation (skin).

### INTERNET PROTOCOL STACKS

These are readily available. Mainly from the U.S. The quality of these, however, varies quite considerably. It is important to purchase the components from a good company. Being able to try the software before purchasing is also very useful. Like any area of componentisation the integration side becomes a significant cost area. As such, this can be minimised by buying components from suppliers that supply a large range of components in the area that you are buying in. For IP this may mean not only IP, ICMP, IGMP, UDP and TCP but also FTP, HTTP and possibly NFS. It is also important to look for how the packets are being passed up the stack - pointers passed or packets copied - as this greatly effects speed of operation. Also provision of facilities such as asynchronous communication or BSD, which is synchronous and is prone to locking up (this problem has to be handled).

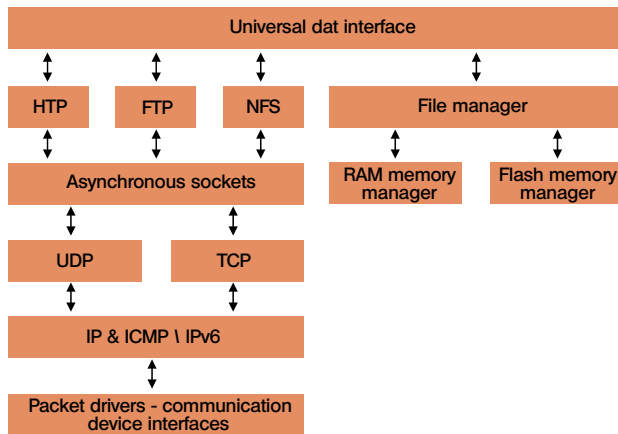


Figure 2.

## ARCHITECTURAL RELATIONSHIP

(See figure 2)

### OPERATING SYSTEMS

Operating Systems (RTOS's etc.) are well catered for in the embedded software components market. Care should be taken, however, to ensure that the application can be separated from the hardware specific aspects of the platform. The best method of doing this, if required, is to use techniques such as configuration enumeration's, configuration files and to use macros for OS calls such as memory allocation calls. These are mainly supplied by U.S. and Canadian companies.

### GSM

There are specialist component suppliers providing the basic building blocks for GSM phones. These are mainly supplied by U.K. companies.

### DIGITAL T.V. SET TOP BOXES

There are specialist component suppliers providing the building blocks for Digital T.V. Set Top Boxes. Care should be taken to ensure that the modules are of good quality and are not costly to maintain and update. Always make sure that you can try before you buy as a wrong decision can be very costly. This is particularly true for browsers with regards to quality and reliability and integration and updating costs. These are mainly supplied by U.K. companies. Browsers are available from a number of U.S. companies.

### COMPONENT LICENSES

The component supplier has to maintain a viable business in order to be able to continue to supply a product to the industry. As such the component supplier maintains the copyright of the components and the purchaser has to sign an agreement acknowledging this. Copyright issues of software components are actually no different than hardware components. With hardware - you own the copyright of the circuit design but the chip manufacturers' own the copyrights of the individual chips.

Component licenses are basically four types:

1. Single platform - this covers a single hardware

design and future versions (new product releases) of that design. This is basically a discounted entry into the component purchase.

2. Site or company wide license - this allows the components to be used on any platforms within the license.

3. Software component license - this covers the situation where the components are used as part of an associated or encapsulating software module. The software module being marketed as a component (not as an end product). The purpose of this license is to ensure that royalty payments are received from what is effectively a competitive business operation.

4. Hardware component license - this covers the situation where the components are used as part of a hardware module. The hardware module being marketed as a component (not as an end product). The purpose of this license is to ensure that royalty payments are received from what is effectively a competitive business operation.

### SUPPORT

Support for the first year is normally supplied with the components. Optional support is available for subsequent years. Care should be taken to ensure that, if updates are required for subsequent years, support is taken out. The support does not, however, refer to the integration of the components into your build. This is a separate issue and has to be handled as a separate cost factor.

### COMPONENTS AND MODULES

Generally when buying components one is buying a set of components that make up a module. It is this module for which a usage license is being purchased. The componentisation of the module allows the component supplier to supply a quality product that can be easily maintained and updated. Care should be taken when buying a module to ensure that it is made up of components. Failure to do so may result in the purchase of a product that is on the edge of failure - ie. changes in one part of the module have ramifications in other parts of the module. These ramifications not being taken into account when the changes are made. Further it will very likely be very difficult and hence very costly to integrate and upgrade.

Good components can be a dream to use. Bad components can be a nightmare - like opening up a can of worms.

The embedded software component industry is relatively new but has a lot of potential. It goes a long way to solve the shortage of skills within the industry. As such using components and supporting, buy purchasing from, component suppliers ultimately has strong benefits for the industry.

It is not unusual for project managers to be unfamiliar with just what can be achieved by using components and by using new techniques used by those components. It is also not unusual for project managers to be unfamiliar with just how much of their project can be

componentised. Often projects are thought of in terms of being company and product specific. The fact is a large portion of programs can be divided into components.

A typical fully componentised implementation would look like this:

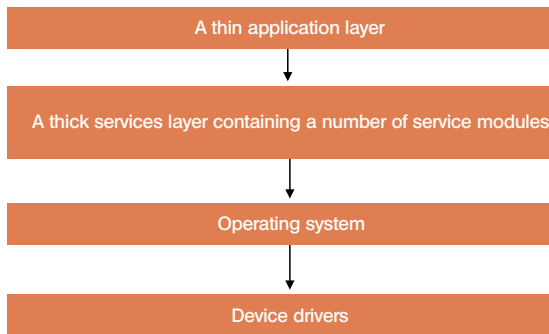


Figure 3.

## EVALUATING COMPONENT SUPPLIERS

When evaluating a component supplier it is important to ask the following questions:

1. does the supplier have a web site and does it supply full information on the components? Alternatively can the supplier supply full written information on the components? Bear in mind that it is very likely that the less information that the supplier is prepared to provide the more that the supplier is trying to hide.
2. does the supplier make available a demonstration version of the components - preferably running within a P.C. build - that can be used for evaluating the components. This is not always possible but where it is possible it should be expected. If a supplier does not supply, for example on their web site, a demonstration version of their components the question has to be asked " Why don't they? ". With some suppliers they have a very good reputation so don't have to. Other suppliers, however, are supplying poor quality modules and as such are using strong marketing techniques to make the sales then, for example, charging you large amounts in support.
3. has the supplier used good techniques to construct the components? This covers a lot of areas such as:
  - i coding standards - important for maintainability and general logical layout of the code.
  - ii general technical issues such as speed of operation - can be critical with IP components.
  - iii specific techniques such as vectoring instead of case statements (faster), asynchronous operation (avoids lock ups) etc..
4. draw up a set of criteria that you are going to use to evaluate the component supplier. Ensure that the criteria are objective. Apply the criteria in your eval-

uation. If the company is sliding all over the place avoid it like the plague - there is a good chance that they don't really have a clue about what your criteria really mean.

5. if possible - look at the general culture of the company and of the individuals within the company as this will have a direct effect on the quality of the code produced. It is very important to look at what you want product wise then work logically stage by stage back to work out what type of company produces this quality of product. It is a very logical process and every stage must be justified and connected. It is very important to have good quality and objective judgement when making your evaluation. A good dose of common sense in conjunction with a good knowledge of life goes a long way. I would strongly urge - avoid politics and avoid fads as they will cost you a lot of money. Generally speaking you will get the best quality and service from small highly motivated and highly professional companies. However there are exceptions as far as the link with size is concerned - small companies can be very poor and large companies can be very good. As always be careful and check in detail.

## CONCLUSIONS

Providing quality suppliers are used - using embedded software components provides the most cost effective and time effective way of developing embedded projects. Using embedded software components also provides significant competitive advantages in terms of ensuring quality and advanced features ■

---

*Kim Lyon is Managing Director of World Computers Ltd., a U.K. based company developing and supplying embedded software modules and components and contract engineering services. He obtained a degree in engineering from Curtin University in Perth, Western Australia. He has had 18 years of broad experience in the industry. Much of this has been in embedded software. He has worked for a number of U.K., U.S., Belgian - German, Japanese and Australian international and multinational companies and for a few small companies (including start-ups). He is highly creative and has a number of patents.*