Position Paper

Modularity and Reuse

Amro Farid, CDAC
Francis Hunt, CTM
James Moultrie, CTM

March 2005

Summary

Modularity is a way to make systems easier to maintain, reconfigure and upgrade. Modular systems are built from *modules* which interact through agreed unchanging *interfaces.* In well-designed modular systems, the required changes will fall within modules. Designing these changes is easier since the designer need only consider the module rather than the whole system. Making these changes is easier since the engineer need only replace one module with another. ***Easier changes are less expensive, and so modularity can enable significant cost savings in maintaining, reconfiguring and upgrading systems.***

Reuse is a related approach, built on modularity and standardisation. ***Reusing components and architectures across ranges of products enables significant cost savings.*** These cost savings emerge through reduced design work, through economies of scale in production and through economies of knowledge when servicing and supporting products. To reuse components across ranges of products, the systems must be modular and their interfaces must be standardised.

Modularity and reuse ideas are not limited to products. They can be applied to production systems and also to work organisations. Modularity itself is a key idea in managing any complex system. Analysis is required to determine where the ideas of modularity and reuse can most usefully be applied since their benefits are not free. Typically modular systems may have extra costs due to the interfaces, and designing reusable components is harder since it involves coping with a much wider set of requirements, and may involve compromises. But by choosing the applications of modularity and reuse carefully, the potential cost benefits indicated above can be made real.

Introduction

Change is one of the few constants in modern life. No longer do we wait until parts are worn out to change them. We change them because technology has improved and we can now do more. Or we change them because we want to configure the system for a new situation or for a new customer. Change driven by these desires for technological progress, configurability and customisation is evident throughout society. It may be hard to predict the technologies of the future, but it is easy to predict that systems will need to change as a consequence. Similarly the desire to configure systems at will, to make the system near optimal in every situation, to do more with less, is a desire that will always be driving systems to change.

Another trend is towards increasing complexity. To deliver all the functionality that a customer wants, product systems need to be more complex. For example, mobile phones have become considerably more complex in order to deliver richer functionality. The latest mobile phones not only include the functionality to make phone calls, but also incorporate cameras, and include software for diary management and finger print recognition.

Change and complexity do not mix well, or at least they do not mix well if we want the change to have predictable results. When we change part of a system we want to have confidence that the system will still function correctly as a whole. We want to know that the change in one part of the system will not have unforeseen consequences in other parts of the system.

A good solution to this problem of changing complex systems is to make the systems *modular*. Modularity is a way to make complex systems understandable and so to enable us to change them with confidence. And being able to change them enables them to achieve near optimal performance over time.

A modular system is one that is composed of *modules* that interact through agreed *interfaces*. There are functional requirements placed on the modules, but how they achieve that functionality is hidden from the rest of the system. Thus the means for achieving a particular function can change over time without affecting the rest of the system, as long as the required functionality is still present and the interface is not changed. A module ideally has minimal dependencies on other modules in the system. This reduces the size and complexity of the interfaces between modules.

The fact that modules are defined through their functional specification and through their interface means that the modules can be designed, produced and tested separately. If they can be developed separately in this way, then they can be developed concurrently and hence there are potential gains in development time. It also means that the organisation designing, producing and testing modules can itself be similarly modular.

If the modules can be standardised then there is potential for systematic reuse, with possibilities of enormous gains in efficiency. For example, rather than designing a different power supply unit for each product in a range, if it is possible to design one to suit them all, then there are obvious savings in design time. There are also savings from economies of scale in production. Finally there are savings in service and support, since engineers only need to be familiar with one power supply unit and therefore transport fewer tools and fewer replacement parts.

These gains from standardised reusable modules do not come for free. In the power supply example above, it will be more difficult to design a power supply unit suitable for all products in a range. The power supply unit will be over-specified for the lower end products and hence more expensive than strictly necessary. Analysis is required to determine what sub-range of products can be served by a common power supply unit.

Reuse ideas can be applied to other things besides modules. Many artefacts can be standardised and hence reused. Many involve reuse of information, rather than the physical artefact e.g. the reusable modules described above are from reusing the design to create identical modules, rather than reusing one physical module in multiple situations.

In summary, modularity enables predictable change in complex systems. Reuse exploits modularity to achieve cost savings, and enables a greater variety of products from the same resource investment. In the remainder of this paper we expand and develop these ideas. The next section identifies the key themes in reuse and modularity. We then identify key research centres and go on to describe a possible future research agenda. We finish with a review of the academic literature on the subjects of modularity and reuse.

Key themes

In the introduction we have explained that modularity is a way of effectively managing change in complex systems. We have explained that reuse is a systematic way of leveraging effort so that work done can be used in multiple settings. In this section we will expand on the key themes contained within the ideas of modularity and reuse. These ideas are further developed in the literature review at the end of this paper, and the appropriate academic references are included there.

Defining modularity and reuse

Modularity is a property of the architecture of a system. An intuitive definition says a system architecture is modular when parts of it, the modules, can be handled separately from the others. Making this more precise, a product architecture consists of three elements:

1. an arrangement of functional elements
2. a mapping from functional elements to physical components
3. a specification of the interfaces between the physical components

An architecture is ***modular*** if it contains

- a one-to-one mapping from functional elements to physical components
- clearly interfaces that decouple the components (the *modules*)

Engineering ***reuse*** can be defined as the business strategy of using existing technological assets that a company controls in the creation of new assets. It can be applied at many levels in a firm, and to many different types of asset. Reuse is a an obvious concept, but can be made more effective by systematic planning. Planning around modular architectures is a key element, since it is likely that some module will be reusable in another setting, even when it is not possible to reuse the whole system.

Advantages and disadvantages of modularity and reuse

The key advantages of a modular architecture are:

- ease of understanding
- ease of change (in particular maintenance, upgrades and reconfiguration)
- independent development
- opportunity for systematic reuse

Reuse has obvious benefits in achieving more with the same amount of resources.

The key disadvantages are:

- a modular architecture may not deliver optimal performance
- designing a reusable modular architecture is difficult and necessitates an upfront investment in time and money with the hope of achieving much greater gains later
- reuse may lead customers to mistakenly believe systems are outdated

Modular architectures can lead to changes in the structure of the business, since they enable suppliers to produce and sell modules. This can be either an advantage or a disadvantage, depending on the strategic plans of the company producing the modular architecture.

The decisions as to whether an architecture should be modular and whether it should be made reusable are essentially economic rather than philosophical. The question is how the desired level of performance, maintainability and upgradability can be most economically achieved. A further consideration is whether the corresponding pattern of financial investment and return is desirable.

Modularity and reuse in product design and development

A distinction is commonly made between development for reuse and development with reuse. A key idea often used when developing for reuse, is that of ***product platforms*** or product families. The use of product platforms enables companies to generate a continuous stream of new products, essential for the success of most companies. A particularly famous example of this is the Sony Walkman, originally launched in 1979, which then spawned almost 250 new models released in the US market during the 1980s. Planning for product platforms can be achieved using product development maps and variants of technology roadmaps.

Development with reuse can opportunistic or planned, but requires a change in mindset of the designer. There are conceptual tools to support development with reuse, such as Clausing's reusability matrix (Clausing 1994). Unilever have successfully exploited development with reuse when designing new manufacturing facilities to achieve a six-fold reduction in the time required to get a reliable estimate of the cost of the facility.

An advanced type of product modularity is seen in modern computer peripherals. Not only can they be plugged in easily, but the computer will reconfigure itself to their presence, an ability known as plug-and-play. This enables unskilled operators to connect new computer peripherals. Applying this idea more widely could have major benefits for the service and support of other equipment.

Modularity and reuse in software engineering

Software has always been a prime candidate for modularity and reuse. This is for three reasons:

- the complexity of software has meant that modularity is the standard design technique
- the benefits of reuse are particularly great since software can be replicated for virtually no cost
- the interfaces can be precisely specified and have no need to contend with issues such as vibration, heat transfer or structural support.

Historically there has been a trend towards reuse in software. Examples of this are high-level platform independent languages such as FORTRAN and C, software libraries, object-oriented programming, design patterns, component-based design and agent-oriented programming. The field of software reuse is mature and many authors have suggested processes that can be incorporated in the software development lifecycle to enable software reuse in companies.

Modularity and reconfigurability in production systems

There has been a drive towards agility and responsiveness in production systems. This is so they can operate more effectively in a dynamic environment and also to enable them to deliver mass customisation. One route to reconfigurable production systems is to build them out of modular mechanical units. In order to exploit reconfigurable production systems, the planning and control software must be similarly flexible, and there has been a corresponding interest in distributed control. The field of holonic manufacturing systems combines these interests in flexible production systems and flexible control, by creating self-contained manufacturing units able to interact, negotiate and resolve problems to achieve optimal production. In this way, the field attempts to realize "plug and produce" production.

Modular organisations

A modular product architecture enables the organisation developing, producing and supporting it to be similarly modular. This has corresponding advantages in enabling the organisation to respond to change. However there are concerns about control of a modular organisation, particularly if different "modules" are under the control of different companies. There is also the danger that important unstructured technical dialogue will be impeded by the interfaces between the organisational "modules".

Key research centres

Modularity and reuse is a broad area, as is made clear in the literature review. Some of the key research centres of which we are aware are picked out below:

- MIT – Center for Innovation in Product Development: Clausing (retired) on design reuse, Whitman on difficulty of reuse in high powered & assembly systems. Eppinger on modularity in product architecture.
- Harvard Business School – Baldwin and Clark on modularity in design.
- Copenhagen Business School – Mikkola on modularity in products and in supply chains.
- Loughborough University – Integrated Product-Process Group, design modularity
- University of Hannover, Germany (IFW)– Institute of Production Engineering and Machine Tools: Reconfigurable Machines, Manufacturing Control Systems, Supply Networks.
- University of Michigan – Reconfigurable Manufacturing Systems Center: Koren on Reconfigurable Machines
- Czech Technical University – Gerstner Laboratory – Agent Technology Group: Marik & Pechoucek on Agent Technology in Production Planning & Control Systems.
- University of Calgary, Canada – Intelligent Systems Research Group: Brennan on Reconfigurable Manufacturing Control Systems.
- University of KU Leuven, Belgium – Production Engineering, Machine Design and Automation Group: Reconfigurable Machines, Manufacturing Control Systems, Supply Networks.

Possible future research agenda

The field of reconfigurable manufacturing systems remains one of most challenging applications for modularity and reuse – especially in response to dynamic conditions in the market and the factory floor. At the machine and manufacturing processes level, much work continues in reconfigurable machines that use incremental and highly flexible manufacturing processes.

In manufacturing system control, there exists much work in the application of agents as a distributed control strategy that supports modular production planning and control solutions. This includes the planning, scheduling, and execution control functions that would support mass-customization. It also includes the shop floor monitoring and diagnosis that is required to make the control solution truly responsive.

The work in distributed control technologies and reconfigurable machines finds a synergy in the field of Holonic Manufacturing Systems. The union of reconfigurable machines and distributed control supports not just mass-customization but also the ability to a manufacturing system composed of "plug and produce" modular units.

Another area that is underdeveloped is tools for developing the business cases for modular architectures, particularly ones that enable the trade-offs between performance, cost and payback period to be made. Case studies of successful and unsuccessful platform approaches would also undoubtably prove very useful, yielding valuable lessons for designers and managers alike.

Key literature

The concepts of complexity, modularity, reconfigurability and reuse are highly intertwined in the academic literature on product development, software engineering, production systems, and virtual enterprises. A survey of the literature yields the following general conclusions:

- modular systems are easier to understand
- modular systems are easier to change
- modular systems enable systematic reuse

The key drawbacks are that:

- modular systems with reusable architectures are harder to design
- modular systems rarely yield optimal performance

In this literature review, we first define modularity. We then look in more detail at the advantages and drawbacks of modularity. We then consider specific considerations in the fields of product development, software engineering, production systems and virtual enterprises.

Defining modularity and reuse

A number of authors provide definitions of modularity (Ulrich 1995; Galvin 1999; Baldwin and Clark 2000; Gershenson, Prasad et al. 2003). We adopt that of Ulrich (1995). In his paper he provides a conceptual analysis of product modularity and its impact on product change, product variety, component standardization, product performance and product development management. He defines a product architecture to consist of three elements:

4. an arrangement of functional elements
5. a mapping from functional elements to physical components
6. the specification of the interfaces among the interacting physical components

He defines an architecture as ***modular*** if it contains a one-to-one mapping from functional elements to physical components with clear interfaces that decouple the components. An architecture is ***integral*** if it contains non one-to-one mappings. These definitions are illustrated in Figures 1 and 2 below.

It is rarely possible for an architecture to be completely modular. For example, if there is a functional requirement on the aerodynamic drag of the total system, then this cannot be mapped to just one of the physical elements. Hence modularity is actually a matter of degree, with some architectures being more modular than others.

“Reuse is an obvious but imprecise concept” (Busby 1998). We adopt the definition from Hunt, Farrukh and Phaal (2001): engineering reuse is the business strategy of using existing technological assets that a company controls in the creation of new assets. It can be applied at many levels in a firm, and to many different types of asset. Modular architectures are a key enabler, since it is likely that some module will be reusable in another setting, even when it is not possible to reuse the whole system.

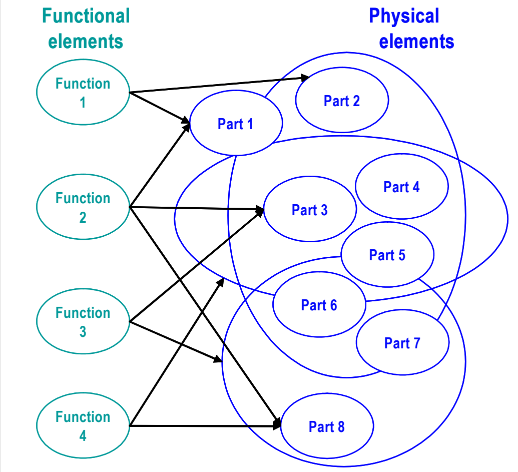

Figure 2: An integral product architecture

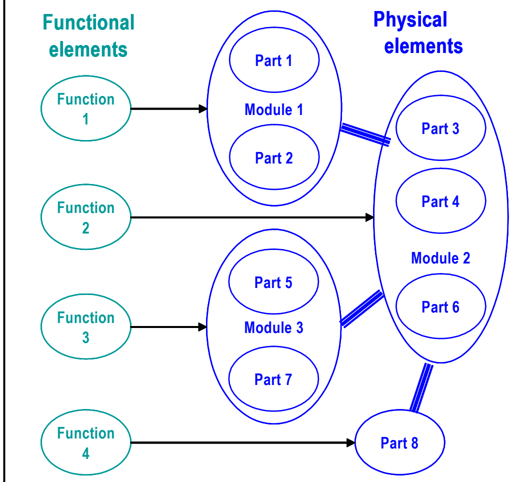

Figure 2: A modular product architecture

It is rarely possible for an architecture to be completely modular. For example, if there is a functional requirement on the aerodynamic drag of the total system, then this cannot be mapped to just one of the physical elements. Hence modularity is actually a matter of degree, with some architectures being more modular than others.

Advantages of modularity and reuse

The fact that a system has been separated into modules confers a number of advantages: ease of understanding, ease of change, independent development and the opportunity for systematic reuse.

We first consider ease of understanding the system, the reduction of the conceptual complexity. Complexity can be defined in a number of ways. Suh (1990) defines the complexity of a design as the amount of information required to adjust a particular system function. Clearly by dividing a system into modules, this type of complexity is reduced. This reduction in conceptual complexity has a number of positive effects. In particular it makes it easier to check that all system functions have been implemented and most importantly it makes it easier to change the system in a predictable way.

Ease of change is perhaps the most fundamental benefit of modularity. Change can occur for a number of reasons, and all can be helped by the use of a modular architecture. Examples are:

- upgrades: new technology (e.g. more memory in a PC)
- add-ons: sell basic unit, provide extras (e.g. bicycle)
- adaptation: different markets or environments (e.g. power supplies)
- wear / maintenance: replacement, fixing (e.g. razor blades)
- consumption: information, energy, materials (e.g. oil, ink, film)
- flexibility in use: user configuration (e.g. camera lens)

By making a system modular, a module can be replaced as long as the replacement module has the same interface. The rest of the system will not be disturbed.

Independent development is another key benefit of modularity. Once the interfaces to a module have been specified, the development of the module can proceed independently and concurrently with other modules. It is possible to experiment with different designs of the module, so long as the interface remains unchanged. Baldwin and Clark (2000) refer to this independent development as *modularity-in-production*.

In addition to ease of understanding, ease of change and the opportunity for independent development, a further benefit of modularity is the opportunity for systematic reuse. This can also be seen as modularity enabling product variety. The essence of this benefit is the possibility of doing more with less, since components and architectures are systematically reused either across a product range, or over time.

To achieve systematic reuse, standardisation is needed in addition to modularity. The interfaces to the modules need to be standardised so that the modules can be reused across a product range. The architecture also needs to be similarly standardised. This reuse of standardised architectures and modules across a product range is often referred to as a product platform or product family approach (Meyer and Utterback 1993; Meyer and Zack 1996; Meyer and Lehnerd 1997; Robertson and Ulrich 1998). The use of product platforms enables companies to generate a continuous stream of new products, a need highlighted by Wheelwright and Clark (1992). A particularly famous example of this is the Sony Walkman, originally launched in 1979, which then spawned almost 250 new models released in the US market during the 1980s.

Standardisation brings in its train other benefits, mainly to do with economies of scale in production and in later service and support. It is a key element of the strategies of design for manufacture and design for assembly.

Pros of modular architectures	Cons of modular architectures
• improves ability to change after introduction • improves variety and speed of introduction of new products • improves maintainability and serviceability • allows development tasks to be decoupled • permits testing of subsystems through their interfaces	• can make products look too much alike • easier for competitive imitations • reduces product performance • more expensive than integral solutions

Table 1: Pros and cons of modular architectures (Cutherell 1996)

Disadvantages of modularity and reuse

An obvious question to ask is why are not all systems modular. There are a number of answers to this.

Firstly, to achieve optimal performance an integral architecture is often better than a modular one (Cutherell 1996). For example, to reduce the weight of a physical system we may assign multiple functions to the same physical module. Or in a computer system to achieve speed, we may allow a program to directly access graphics hardware rather than use the standard interface. In general to take a system to the limit of performance, we may need to overload modules with multiple function and we may need to bypass or do away with standard interfaces. This will have costs in terms of maintainability and ability to change, so we achieve optimal performance at a particular point in time at the expense of near-optimal performance over an extended period.

A similar argument applies to the use of product platforms. Since the platform is to be usable across a range, at the lower end of the range parts of the platform are likely to be over engineered. Thus the unit cost is likely to be higher than necessary at the lower end of the range. There is a trade-off in this case between optimal cost at a particular point in the product range, against near-optimal cost across the entire product range.

A second difficulty comes in designing systems to be reusable either across existing product ranges or over time. This is a difficult task. It involves handling more requirements and constraints than if the system were to be used in just one setting. It also involves predicting the future. The designer has to be able to predict the likely the changes needed in the future and ensure that the locus of these changes is in the modules and not embedded in the architecture.

Finally there are business considerations which may weigh against modularity. Modularity and standard interfaces may allow a user to mix and match offerings from different companies. This is beneficial to the user, but is not necessarily beneficial to some of the companies brought into competition. It may be beneficial if it allows a company to refocus on its core competences and use external suppliers for non-core elements. A further business consideration related to the use of modular architectures is that although evolutionary change becomes much easier, revolutionary changes involving change of architecture may become much harder, particularly if the architecture has become embedded in the company organisation (Henderson and Clark 1990).

As has been seen, modular architectures have advantages and disadvantages. The decisions as to whether an architecture should be modular and whether it should be made reusable are essentially economic rather than philosophical. The question is how the desired level of performance, maintainability and upgradability can be most economically achieved. A further consideration is whether the corresponding pattern of financial investment, with a large initial upfront investment and a large but delayed return, is desirable.

Modularity and reuse in product design and development

Many of the issues relating to modularity and reuse in product design and development have already been mentioned in the preceding sections This is because modularity ideas are most often associated with product modularity. In this section we draw out some of the implications.

First there is a key distinction between ***development for reuse*** and ***development with reuse***. Development for reuse uses the product platform and product family ideas described above. It involves a major upfront investment in time and money to produce the platform, then the benefits are reaped later from the ease of developing derivative versions. Planning for product platforms can be performed using product development maps (Wheelwright and Sasser 1986; Meyer and Utterback 1993; Meyer and Zack 1996; Robertson and Ulrich 1998). Another planning tool that aims to integrate product strategy, technology strategy and competences strategy is Andrade and Clausing's (1997) Business Integration Map, essentially a technology roadmap (Phaal, Farrukh et al. 2004). A general process for addressing systematic reuse is outlined in (Hunt, Farrukh et al. 2001).

Development with reuse is a strategy of reusing existing components and architectures when developing new ones. It is a natural strategy and does not require the components to have been developed with reuse in mind. However upfront investments in reusable components and architectures enable most benefit to be achieved. One conceptual tool for encouraging and supporting designers in development with reuse is Clausing's (1994) reusability matrix. This encourages the designer to look for sources of reusable ideas, whether in current products, competitor products or analogous products. It applies this at multiple levels within the product architecture.

When backed by systematic generation of reusable components, the strategy of development with reuse can be particularly effective. Unilever populated a database with objects representing standard processing units in their production plants. By assembling the design for new plants from these reusable components, they managed to reduce the time to obtain a cost estimate for new plant from six weeks to one week (Milner and McFarlane 1998). They could also involve plant managers in many of the design decisions.

Ulrich (1995) identifies three types of modular architectures. In slot architectures modules of different types have different interfaces. For example, in a car a module such as the car radio will have a slot that a car radio can fit in, but the slot would not accommodate a module such as the wing mirror. In a bus type architecture all modules have the same type of interface and connect to a common bus. A good example of this is USB on computer systems, where all USB devices connect to a common bus. A final type of modular architecture is sectional, where all modules have the same interface but there is no common module to which all modules connect. Examples of this are piping systems. These are illustrated in Figure 3.

Modern computer peripherals illustrate an advanced type of modularity. Not only can they be plugged in easily, but the computer will reconfigure itself to their presence, an ability known as plug-and-play. This enables unskilled operators to connect and use computer peripherals, and has major implications for service and support.

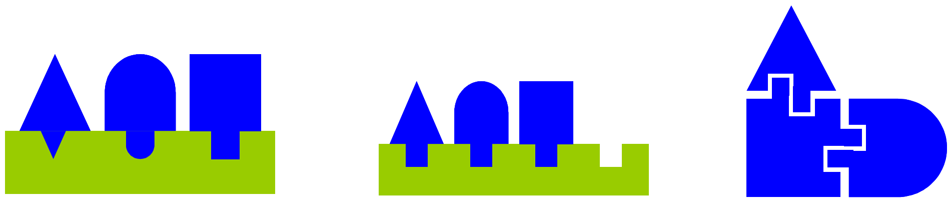

Figure 3: Various types of modular architecture (Ulrich 1995)

Modularity and reuse in software engineering

Software has always been a prime candidate for modularity and reuse. This is for three reasons: the complexity of software has meant that modularity is the standard design technique; the benefits of reuse are particularly great since software can be replicated for virtually no cost; and the interfaces can be precisely specified and have no need to contend with issues such as vibration, heat transfer or structural support.

Historically there has been a trend towards reuse in software. The development of higher level languages such as FORTRAN and C, was in part due to a desire to make programs reusable across different operating systems. Furthermore code that did particular common tasks was grouped into libraries of functions that could be used again and again. Thinking on modularity developed and it was realised that a natural modularisation should include not only the functions associated with a module, but also the data structures. This gave rise to object-oriented programming.

Component-based design is an extension of the object-oriented model. In this the designer can take a component such as a desktop window and set particular properties, such as in this example the title, the background colour, and whether it has a scroll-bar. This enables rapid construction of software. Applying the plug and play model to software also suggests attractive ideas:- the idea of software modules identifying the presence of other software modules and configuring themselves accordingly. This gives rise to the idea of software agents and agent-oriented programming. A further example of reuse in software, at the level of design, is the software pattern movement. In this, object oriented designs that are particularly effective have been catalogued in the public domain for other designers to draw upon.

Much has been written on software reuse. Poulin (1999) and Yongbeom and Stohr (1998) both survey the field. In particular, small libraries of high quality components are more useful large libraries of moderately good components. A number of books exist advising on how software reuse should be implemented (Karlsson 1995; Rada 1995; Jacobson, Griss et al. 1997; Poulin 1997; Reifer 1997). In general these scope and define software reuse; propose a generic software reuse process and indicate how this can be incorporated into the software development lifecycle; and explore the issues that the authors believe will have an impact on the success of the software reuse program.

Yongbeom and Stohr's (1998) process consists of the two broad activities of producing reusable components (identifying opportunities, classifying components) and consuming reusable resources (retrieval, understanding, modification and integration). Rada (1995) offers a more detailed breakdown of activities and adds a third broad activity of managing the reusable components.

The issues that the authors consider as important include technical issues, organisational issues (senior management support, not invented here syndrome), economic issues (cost benefit analyses) and legal issues.

Modularity and reconfigurability in production systems

As the pace of the manufacturing world has increased, and customers have become more demanding, manufacturing firms have had to become increasingly agile and responsive. This has often had its implications on the modularity and reconfigurability of all parts of the shop floor. Flexible manufacturing systems (FMS) and single minute exchange of dyes (SMED) focused primarily on the interchangeability of tooling as the modular unit (Black 1991). Later, the concept of reconfigurable machines and transfer lines was introduced to facilitate the usage modular machining units with quick installation times to the factory floor.

Taking a different approach, researchers of software agents have exploited the reconfigurability and modularity of software to improve the responsiveness of shop-floor production planning and control (Shen 2001). A key use is for monitoring and diagnostics of operations. In a sense, agents could be used as "smart-sensors" of the shop-floor operations to detect and reconfigure for special conditions like machine breakdowns and rush-orders (Heikkila 1997). Equally as important the multi-agent systems have been applied to planning and execution as a strategy towards achieving the shop-floor control necessary for mass customization.

The field of Holonic Manufacturing Systems tries to combine these two trends of reconfigurability in machine and software. In this regard, the field attempts to design manufacturing systems from modular manufacturing units composed of their own reconfigurable machine and production planning and execution control software. In this way, the field attempts to realise “plug and produce” production (McFarlane and Busssman 2003).

Modular organisations

A modular product architecture enables the organisation developing, producing and supporting it to be similarly modular (Sanchez and Mahoney 1996). Sanchez and Mahoney propose that this sort of organisation can respond more flexibly to environmental change. As an example of this, Sturgeon (2002) proposes a new organisational paradigm in modular production networks.

Chesborough and Teece (1996) examine when such virtual organisations are virtuous and highlight the need to avoid opportunistic behaviour when operating as a loosely coupled network of organisations. Large integrated organisations may adapt less rapidly, but they do have mechanisms for control and for sorting out conflicts. Other arguments that may weigh against modular organisations centre round the need for unstructured technical dialogue (Monteverde 1995), the presence of “sticky information” (von Hippel 1994) or tacit information exchange (Afuah 2001).

Bibliography

Afuah, A. (2001). "Dynamic boundaries of the firm: Are firms better off being vertically integrated in the face of a technological change?" Academy of Management Journal **44**(6): 1211-1228.

Andrade, R. and D. Clausing (1997). Strategic integration of products, technologies and core competences. Proceedings of the International Conference on Engineering Design: 554-558.

Baldwin, C. Y. and K. B. Clark (2000). "Managing in an age of modularity." Harvard Business Review **75**(5): 84-93.

Black, J. T. (1991). The Design of the Factory with a Future. New York, McGraw-Hill.

Busby, J. S. (1998). Causal explanations of the absence of reuse in engineering design organisation. Proceedings of the Engineering Design Conference '98: Design Reuse. S. Sivaloganathan and T. M. M. Shahin.

Chesbrough, H. W. and D. J. Teece (1996). "When is virtual virtuous? Organizing for innovation." Harvard Business Review **74**(1): 65-&.

Clausing, D. P. (1994). Total quality development. New York, ASME Press.

Cutherell, D. (1996). Product Architecture. The PDMA Handbook of New Product Development: 217-235.

Galvin, P. (1999). "Product modularity, information structures and the diffusion of innovation." International Journal of Technology Management **17**(5): 467-479.

Gershenson, J. K., G. J. Prasad, et al. (2003). "Product modularity: definitions and benefits." Journal of Engineering Design **14**(3): 295-313.

Heikkila, T. (1997). "Holonic Control for Manufacturing Systems: Functional Design of a Manufacturing Robot Cell." Integrated Computer-Aided Engineering **4**: 202-218.

Henderson, R. M. and K. B. Clark (1990). "Architectural innovation: the reconfiguration of existing product technologies and the failure of established firms." Administrative Science Quarterly **35**: 9-30.

Hunt, F. H., C. J. Farrukh, et al. (2001). Technology reuse: developing a practical approach to making the most of your technological assets. Proceedings of IAMOT 2001. Lausanne.

Jacobson, I., M. Griss, et al. (1997). Software reuse: architecture, process and organisation for business success. Reading, MA., Addison-Wesley.

Karlsson, E.-A. (1995). Software reuse: a holistic approach. Chichester, Wiley.

McFarlane, D. C. and S. Busssman (2003). Holonic Manufacturing Control: Rationales, Developments and Open Issues. Agent-Based Manufacturing: Advances in the Holonic Approach. Berlin, Springer-Verlag.

Meyer, M. H. and A. P. Lehnerd (1997). The power of product platforms: building value and cost leadership. New York, Free Press.

Meyer, M. H. and J. M. Utterback (1993). "The product family and the dynamics of core capability." Sloan Management Review **34**: 29-47.

Meyer, M. H. and M. H. Zack (1996). "The design and development of information products." Sloan Management Review **37**: 29-47.

Milner, J. M. and D. C. McFarlane (1998). Modularity in production systems: an approach to system design for a turbulent future, Institute for Manufacturing, University of Cambridge.

Monteverde, K. (1995). "Technical dialogue as an incentive for vertical integration in the semiconductor industry." Management Science **41**(10): 1624-1638.

Phaal, R., C. J. P. Farrukh, et al. (2004). "Customizing roadmapping." Research Technology Management **47**(2).

Poulin, J. S. (1997). Measuring software reuse: principles, practices and economic models. Reading MA, Addison Wesley Longman.

Poulin, J. S. (1999). "Reuse: been there, done that." Communications of the ACM **42**(5): 98-100.

Rada, R. (1995). Software reuse: principles, methodologies, practices. Oxford, Intellect.

Reifer, D. J. (1997). Practical software reuse. Chichester, Wiley.

Robertson, D. and K. Ulrich (1998). "Planning for product platforms." Sloan Management Review **39**(4): 19-+.

Sanchez, R. and J. Mahoney (1996). "Modularity, flexibility and knowledge management in product and organisation design." Strategic Management Journal **17**(Winter special issue): 63-76.

Shen, W. (2001). Multi-Agent Systems for Concurrent Intelligent Design and Manufacturing. London, Taylor and Francis.

Sturgeon, T. (2002). "Modular production networks: a new American model of industrial organization." Industrial and corporate change **11**(3): 451-496.

Suh, N. P. (1990). The principles of design. Oxford, Oxford University Press.

Ulrich, K. (1995). "The role of product architecture in the manufacturing firm." Research Policy **24**(3): 419-440.

von Hippel, E. (1994). ""Sticky information" and the locus of problem solving: implications for innovation." Management Science **40**(4): 429-439.

Wheelwright, S. and K. Clark (1992). "Creating product plans to focus product development." Harvard Business Review **67**(2): 70-82.

Wheelwright, S. and W. Sasser (1986). "The new product development map." Harvard Business Review **62**(3): 112-125.

Yongbeom, K. and E. Stohr (1998). "Software reuse: survey and research directions." Journal of Management Information Systems **14**(4): 133-147.