A MODULAR APPROACH TO THE AIRCRAFT PRODUCT DEVELOPMENT CAPABILITY

Gunnar Holmberg
Future Products, Saab AB
gunnar.holmberg@saab.se

Abstract

This paper[1] discusses some properties of future aerospace and defence systems in terms of need for flexibility and adaptability due to growing complexity of the system and increasing dynamics in its context. Some ideal properties of such a system are given in a general form using an axiomatic design representation.

A framework is proposed to address the management of a product development capability, PDC, for this family of flexible complex systems, starting from the three dimensions: product, process and supply chain, as used in 3D concurrent engineering. Tight couplings exist between the three dimensions that are all time dependent, and inevitable changes due to e.g. obsolescence and changing context need to be managed in the three dimensions.

The complex system is analysed for its classes of sub- systems from a perspective of flexibility.

1 Introduction

The development of aircraft involves challenges in many aspects. The product, subject to many design drivers, uses a wide range of technologies in highly optimised conditions with ever-increasing cost and time pressure.

Defence systems are used in an increasingly complex context. An increased flexibility in the type of operations is important. New types of interoperability are needed, in order to be able to combine different defence systems and forces, and in order to manage the increasing level of integration required to support the operation.

Like most complex systems, the overall life cycle for defence systems is an order of magnitude longer than for some of the sub-systems it is composed of [1]. This implies that issues like obsolescence and changing requirements are driving a need for an easily upgradable system in order to ensure the required functionality and integrity of the system.

Consortiums for projects are formed with considerations such as risk sharing, market and technology access as well as make-buy-processes creating demanding collaborative situations.

From the above follows that the product development capability, PDC needs to be flexible and adaptable to efficiently consider new possibilities and meet new requirements with minimum delays as well as being able to act in collaborative supply chains of various types.

Other design drivers for the PDC include e.g. ability to reduce risk, maximize robustness, have good integration ability, cost reduction and streamlining development work. The effort to achieve and maintain a PDC is considerable and include people, process and computer support aspects. As in all complex designs there are contradictions between the different design drivers, and the design of the PDC have to be architected from a holistic perspective.

---

[1] A result from the LARP III research program funded by the Swedish Aeronautical Research Program and Saab, performed by Saab, Linkopings Universitet and Ecole des Mines de Paris.

This paper proposes a framework to manage the product functionality through-out its life for a complex system product. Such complex systems could be on different levels e.g. on the level of air traffic system (aircraft in its context), aircraft level, or a sub- system within the aircraft. The framework is recursive and intended for any of these systems levels subject to the conditions given below, and is referred to as a complex system.

Some key aspects for the considered complex system are:

- requirements are changing through- out the life of the system and are not fully known à priori.
- need for adaptability through flexibility
- growing complexity and systems of systems integration
- long life with a wide range of life in the different subsystems it is composed of
- strong need to maintain system integrity over time

First a framework is given, then a discussion on the PDC aspects follows.

2 Framework

A framework for the PDC analysis is given. It is based on axiomatic design [2,3] to represent the requirements, system solutions and their relation for the concerned system family and its evolution throughout the life. It further uses the dimensions of 3D concurrent engineering (3DCE) established by Fine [1] , which is a method to analyze and concurrently develop the aspects of the three dimensions product, process and supply chain emphasizing the time dynamics of the dimensions relating to a harmonic behavior characterized by the clockspeed of the system.

Here, the life cycle is used to describe implications of the long life cycle, aspects and ideal properties of the three dimensions are described for the complex system.

2.1 The life cycle

A company, or a set of partner companies, developing complex systems normally manages their product portfolio by starting from a strategy defining the type of business and products that it intend to have in its future portfolio. In order to position for coming business the company pursue activities including :

- technology acquisition and development to support a target portfolio
- business development
- operational development

in order to ensure that it is sufficiently prepared in terms of technology access, and operational capability when the right business conditions arrive to launch a new product.

The starting point for the life cycle is here defined as the time where it is decided to bring a new product to the market. At this time it is important that all major technologies of the launched project are in place in order to be able to execute an efficient and controllable project with acceptable levels of risk.

2.1.1 The life cycle for a single sub- system

The applicable life for a system could be viewed from a value and cost perspective over time, see figure 1.

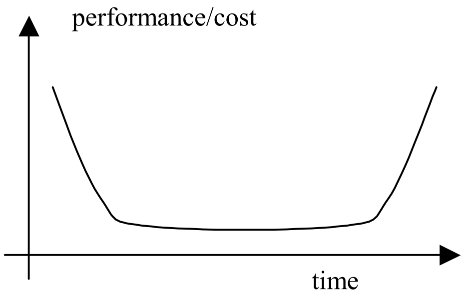

Figure 1: Sub-system performance/cost ratio over time

An early application of a technology, methodology or tool implies low maturity e.g. exposure to instabilities, bugs and difficulties to find skilled implementation guidance and support for its application. Sometimes it has a high acquisition cost for its uniqueness as well. On the other hand early adoption may provide competitive benefits that outweighs the disadvantages in certain strategic cases.

In the following stable, or dominant, phase the technology have reached maturity, it is well understood, several suppliers exist, training is available etcetera.

At the end follows the degradation of the system relevance, the requirements on the system have changed to an extent where the system becomes less and less optimal, it is built on technologies which are out of date, the interfaces may be overtaken, new generations are outperforming the system and so on.

This technology life–cycle applies to single sub-systems, tools and processes and ranges over more than an order of magnitude, e.g. an airframe have a life in the order of 50 years, while some avionics and flying software have a useful life of less than five years. The latter often creating obsolescence problems.

2.1.2 The life cycle of a complex system

For a complex system composed of a number of sub- systems it is necessary to support asynchronous upgrade of the sub- systems while maintaining the integrity of the complex system. The complex system is subsequently subject to an essentially more complex life cycle, it is subject to architectural upgrades, retrofits, upgrade of involved sub-systems which all drive an extensive branching of the life cycle that needs to be handled.

A long life cycle for a complex system is then defined such that the life cycle is long if a set of the sub- systems have to be replaced in order to ensure the functional integrity of the complete system during its life, e.g. due to obsolescence.

In order to reflect this system evolution, initial activities in advance development plays a critical role for the ability to maintain an updated capability through- out the system life.

2.2 The product- the complex system

Adopting the principles of axiomatic design to represent a system, Suh [2,3] describes a system subject to a requirement space represented by functional requirements, FR_i $1<i<M$, and a solution space represented by design parameters, DP_j $1<j<N$, such that a design solution that meets **FR** use **DP** in a combination represented by the design matrix, **DM** with the elements DM_{ij}. The theory of axiomatic design states a number of axioms, corollaries and theorems. Satisfying the Independence and Information axioms together with the theorems gives an ideal design of a system where M equals N (theorem 4) and all elements in DM_{ij} where $i \neq j$ are zero. A system where **FR** and **DP** can be grouped such that those groups are independent from the rest of FRs and DPs is modular, see equation 1.

$$\begin{Bmatrix} FR_1 \\ FR_2 \\ \dots \\ \dots \\ FR_M \end{Bmatrix} = \begin{bmatrix} X\ 0\ \dots\dots\dots \\ 0\ X\ 0\dots\dots. \\ 0\ 0\ X\ 0\ \dots. \\ \dots\dots\dots\dots. \\ \dots\dots\dots 0\ X \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \\ \dots \\ \dots \\ DP_N \end{Bmatrix} \tag{1}$$

Theorem 5 states that if a new functional requirement appears, then a re- design is needed. If the system is scalable in the space where new functionality is added , this FR_{M+1} could be met by addition of a corresponding DP_{N+1}, without influencing the previous design.

Published by the International Council of the Aeronautical Sciences, with permission

For systems where the integration is stronger than in this ideally modular, an architecture is needed to support the modular decomposition. This provides a coupling between the DP:s and is subject to its own FR:s not explicitly stated. Examples of such functionality are

- the coupling between systems functionality, data buses and harnesses
- failure mode monitoring and management, functional degradation
- systems with a strong coupling between different behavior requirements, e.g.
    - flight control system
    - optimized structures such as wings.
    - power supply

The corresponding expansion of this representation is illustrated in equation 2a and 2b. This represents a complex system at a certain time of the life cycle.

$$\left\{\begin{array}{l} FR_1 \\ FR_2 \\ \dots \\ \dots \\ FR_N \end{array}\right\} = \left(\begin{array}{c:l} X & 0 \;\dots\dots\dots \\ X & X\;0\dots\dots \\ X & 0\;X\;0\;\dots. \\ X & 0\dots\dots\dots. \\ \hdashline X & 0\dots\dots 0\;X \end{array}\right) \left\{\begin{array}{l} DP_1 \\ \hdashline DP_2 \\ \dots \\ \dots \\ DP_N \end{array}\right\} \quad (2a)$$

or, including time dependencies and decomposition into an architectural part, $\mathbf{DM}_a$, represented by the coupled first column and corresponding DP_1. The modular functionality is composed of an architectural contribution and modular extensions in the uncoupled $\mathbf{DM}_m$ such that

$$\mathbf{FR}(t)= \mathbf{DM}(t)\;\mathbf{DP}(t)= [\mathbf{DM}_a:\mathbf{DM}_m(t)]\;\{\mathbf{DP}_a(t):\mathbf{DP}_m(t)\} \quad (2b)$$

Linear change of the system exist for a possible given range, such that $\mathbf{DP}_{min} < \mathbf{DP} < \mathbf{DP}_{max}$. However, the complex system has a long life and will be subject to new requirements and new solutions which means that the architecture, $\mathbf{DM}_a$, has to be implemented such that not yet known FR:s and DP:s could later be integrated while $\mathbf{DM}_a$ basically remain invariant, see equation 3, where types of changes are given in bold.

$$\left\{\begin{array}{l} FR_1 \\ \dots \\ FR_i \\ \dots \\ FR_M \\ \mathbf{FR_{M+1}} \end{array}\right\} = \left(\begin{array}{l} X\;0\;\dots\dots\dots \\ X\;X\;0\dots\dots \\ X\;0\;X\;0\;\dots.\mathbf{X}_{i,\,N+1} \\ X\;0\dots\dots\dots. \\ X\;0\dots\dots 0\;X \\ \dots\dots\mathbf{X}_{M+1,\,j}\dots\mathbf{X}_{M+1,\,N+1} \end{array}\right) \left\{\begin{array}{l} DP_1 \\ \dots \\ DP_j \\ \dots \\ DP_N \\ \mathbf{DP_{N+1}} \end{array}\right\} \quad (3)$$

The changes could be from a new FR_{M+1} (met by an existing DP_j or a new DP_{N+1}), or from a new possibility in DP_{N+1} (replacing existing solution for FR_i , e.g. due to obsolescence), or from a combination of the two. As long as $\mathbf{DM}_a$ remain stable those changes are independent of the rest of the design. When $\mathbf{DM}_m$ maintains its diagonal form (when pivoted) the development could be done in full concurrency, however this ideal property have sometimes to be sacrificed and $\mathbf{DM}_m$ could partly be triangular, forcing a certain sequencing in the implementation. An example of this for aircraft is the harness that could not be completed before all signals are identified, but finalizing the harness at architectural level would be non optimal even though the major layout belongs to the architectural phase at current technology level.

This has to be distinguished from a platform and modular approach for e.g. cars where basically the range of FR:s that will apply to a family of cars using the same platform is known in advance.

The architectural part of the system is evolving during the life time of the system and is subject to changes at a rate typically slower than for the sub systems [1].

If the architecture of the complex system fails or is applied outside of its bounds, the system will become complicated, as exemplified in equation 4.

$$\begin{Bmatrix} FR_1 \\ FR_2 \\ \ldots \\ \ldots \\ FR_M \end{Bmatrix} = \begin{bmatrix} X\ 0\ X\ X..X \\ X\ X\ 0\ldots X.. \\ X\ 0\ X\ 0..\ X \\ X\ 0\ \ldots X\ldots. \\ X\ 0\ X\ X\ \ X \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \\ \ldots \\ \ldots \\ DP_N \end{Bmatrix} \quad (4)$$

It is obvious that this type of system is difficult to develop, verify, validate and certify as well as to introduce later changes to.

2.3 The process

The process is the set of methods and tools available to facilitate the execution of projects relating to a particular product. 3DCE usually focuses on the manufacturing process, while in this paper it covers also the development process at all stages of the life cycle and the establishment of a development process in advance development is assumed.

The process shall support the design iterations that are searching for the optimal **DP** to meet **FR**. The iterations are typically rapid at early conceptual stages, changes are implemented quickly and only knowledge produced quickly enough has a possibility to influence the design. At later stages, convergence is the focus and design iterations are slower, focusing on ensuring the consistency of the solutions. In general the major part of the systems capability and cost is defined at early stages.

Typical components of the process are methods and tools for e.g.:

- A life cycle description with its maturity gates and review schemes. (to structure and manage the time dimension)
- Requirements Management (to manage **FR**)
- Configuration Management (to manage **DP** and their fulfilment of **FR** through **DM**)
- Modelling and Simulation (to support the understanding of and optimise **FR**, **DP** and **DM**)
- Generic workflows and descriptions at various levels supporting integration of the system or development of individual sub-systems
- Project Management (to execute the PDC)

Processes have to be designed such that Product development and systems engineering for the complex system supports the establishment of a system design according to the previous chapter. I.e. it has to support the establishment of a decomposition into a stable architecture, $\mathbf{DM}_a$, and modules, $\mathbf{DM}_m$, that provides the flexibility to the system functionality.

In order to achieve this, the development process can be decomposed into one architectural part achieving the architecture and ensuring the integration of the modules into the architecture to provide the full system, while the rest of the development process should support concurrency for the modules development as well as meet the integration requirements.

Furthermore, it has to quickly respond to new FR:s and DP:s as given in equation 3.

2.4 The supply chain

A complex system often have a complex supply chain. Reasons for the design of the supply chain varies, but components are risk sharing, market access, technology access and cost effectiveness.

The rationale for selecting the partnership and supply chain varies over time, where e.g. venture capital and technology access may dominate at an initial stage to later be dominated by sustainability of the system and ability to change. See e.g. the Gripen fighter A/C General Electronics Control Unit case [4] where technology access played a central role at the beginning, while the increase of criticality for being able to change software functionality quickly, later have led to insourcing of certain systems.

Fine [1] describes the logic of outsourcing based on e.g. the double helix, figure 2,

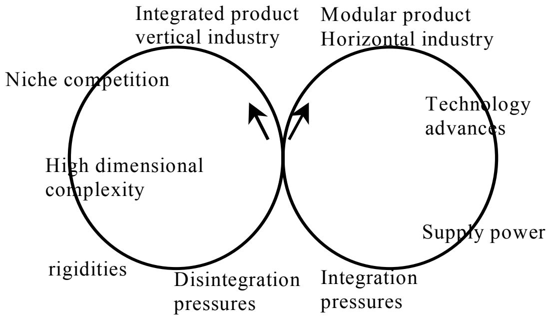

Figure 2: the double helix from Fine [1]

where an industry that is vertically integrated have difficulty to provide sufficiently innovative development of all technologies and sub-systems that the company's products cover. As a consequence the vertical integration is challenged by a possible supply base of companies more focused on one or a few of these sub- systems and a more modular product evolves where horizontal integration dominates and sub- systems develops quickly. This creates risks of supply power and difficulty to maintain the integrity of the system which generates an integration pressure and the supply chain starts to integrate vertically again. This pattern that may have a very long lead time, Fine is estimating a 20 to 30 years cycle for Aerospace industry, covers the supply chain dynamics together with the changing rationale as exemplified above.

For complex systems with long life cycles it follows that not only the new systems will be subject to the changes according to the double helix, but also the existing system instances that will have to maintain integrity and upgraded functionality.

The ideal supply chain, from product flexibility and integrity point of view, will hence need to have interfaces and flexibilities built in that matches the changes to the system as such.

A key aspect for the integrator is to be able to develop and manage architectures such that the supply chain may be chosen flexibly. Some of the sub-systems of e.g. critical importance to the system may remain with the integrator. Sub- systems in general should be distributed in a supply chain that is using a partnering strategy to identify how to integrate or interface with the different suppliers. These priorities will vary over time.

To develop systems in a distributed supply chain is identified as one of the key lead time risks. In particular, when much of the innovation comes from sub- system level as is the case for a complex system that has a dominant design [5]. It is important to establish an efficient and dynamic interaction with the existing and potential/ candidate supply chain.

Modeling and simulation is a key enabler to achieve communication dynamics in the supply chain [6, 7]. It is however important to ensure that model exchange and sustainability do not build lockings into the supply chain with high entrance thresholds. Further, it may lead to expensive change propagation due to tightly coupled models when the supplier is not concerned by the changes. Several levels of interaction is possible ranging from verbal exchange supported by models to integrated shared models with consistent change propagation.

3 Product Development Capability

The product development capability of a company is its ability to use the process and the supply chain to develop and deliver a systems product.

PDC:s have been developed for many years through initiatives in e.g. holistic product development [8], integrated product development [9], simulation based acquisition [10] and systems engineering [11].

The importance of the different dimensions for the PDC depend on the type of situation, e.g. from market conditions classified in [12] as product excellence, customer intimacy and operational excellence. For product excellence (which has been dominating for figther aircraft) the product dimension dominates. For customer intimacy (which was one of the success factors for Gripen [13]), it is a combination of the three dimensions with customer communication as the key. For operational excellence (the model for much of the commercial aircraft business with growing importance for defense systems as well, as affordability is key) the process dimension dominates.

3.1 The complex system in its context

The complex system has a range of flexibilities identifiable for its various subsystems (ability to adapt its DP:s), ranging from rigid to adaptable. The functional requirements of the system are ranging from static to dynamic (FR:s). Sub- systems can from this be positioned in quadrants according to figure 3.

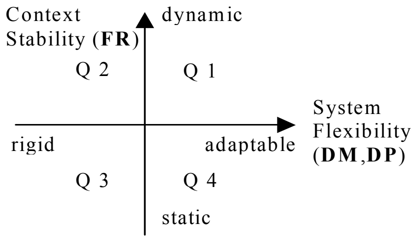

Figure 3: System flexibility and context dynamics

Sub-systems in *the first quadrant, Q1,* are subject to frequent changes in requirements and are adaptable. The adaptability may change over the life cycle, e.g. for a decoupled design that is subject to a coupled and integrated verification, validation and certification (VVC). This will not support the modular entry into service of changes for these systems, as it will inhibit their ability to execute decoupled change projects.

All functionality with high clockspeed have to be designed into this quadrant. Sub-systems in this class being properly supported by the architecture such that the whole systems integrity is not dependent on each subsystem should be supported by development principles such as emergence mechanisms proposed by Highsmith [14]. He proposes to create innovation fields by setting the FR:s enough demanding to get a project situation pushed to the limit of chaos. This is only possible when full system integrity is not at risk, but by decomposing and identify the areas of most expected innovation intensity, this could well be very beneficial for the system adaptability.

In order to cater for efficient and innovative working conditions it is important that a large degree of decision freedom on working conditions for the concerned team is provided. The integrator should basically provided constraints and possibilities required for high quality integration into the architecture. If such a subsystem is provided by a supplier then the minimum interface requirements to ensure systems quality should be enforced while the supplier should have the possibility to optimize their PDC. This further simplifies the interface to the supplier, and allows for the integrator to look for new technologies and suppliers wiht less constraints and thus be able to maintain a more ideal supply chain throughout the life of the system.

Sub-systems in *the third quadrant, Q3,* are referred to as stable sub-systems. Those systems have well defined requirements that are not expected to be changed. Initial effort could be large for these systems in order to optimize their performance as it will contribute to the whole life cycle of the system, as well as the additional effort of integration is not necessary to repeat while changes are small and rare.

Sub-systems in *the second quadrant, Q2,* are pulled by the change of requirements and often become a real problem as they could not respond to the changes in the requirements that will occur. This may endanger the integrity and usefulness of the system. These systems have either to be designed with a higher focus on adaptability (which is likely to reduce their initial level of performance), or be decomposed into more flexible parts corresponding to dynamic requirements expected and into static parts not likely to change. I.e. the system is restructured to decompose into quadrants one and three.

Sub-systems in *the fourth quadrant, Q4,* has an over-capacity in adaptability/ flexibility or a technology development which is pushing the system forward. These systems have to be stabilised in order to avoid cost generating change volumes where they do not serve a purpose for the overall system.

The global systems behaviour could respond well to changes of requirements, I.e. be in Q1, even if many of the sub- systems it is composed of belongs to Q3 given that the architecting have achieved a decomposition that positions the sub- systems that are subject to change in the first quadrant. This provides a system where stability and flexibility are complementary. As pointed out by Murman et.al. [15] based on Utterback [5] a large extent of the changes and innovations could be expected at sub-system levels when the overall system is in a dominant design phase which should provide the basis for the proposed type of architecture. On the other hand, if the architecture fails to place the critical systems in the first quadrant, or fails to achieve modularity, then the global system is at risk to end up in the third quadrant with very limited possibility to respond to changes.

From the above it can be seen that neither a homogenous process nor a homogenous organisation culture could be expected to support the management of such a system development and sustainment. The ideal conditions for sub- systems in Q3 are very different from those in Q1. One should rather strive for certain general conditions that gives the right integrating conditions for various micro cultures in the company and throughout the supply chain. This enables the possibility to optimize with e.g. local influence as long as consequences are local.

3.2 Architecture for a long life cycle

As have been described above, a complex system like an aircraft with a long life cycle will be subject to changes in its context and content that will not be possible to reflect in the initial design specification. If the architecture is developed together with the first generation of the product, then there is a clear conflict between the long term aspects of the architectural solution needed and the lack of time for achieving the first release of the system.

The obvious consequence is that a project that handles both the architecture and the first release of the system will always tend to trade away from long term optimal solutions as those are less precise and the consequences of not meeting them less obvious, hence more difficult to defend. I.e. the set of FR:s for the first release of the product will have an over- emphasized influence on the system.

A product development should therefore be divided into an architectural phase where the architecture, $\mathbf{DM}_a$ and $\mathbf{DP}_a$, is developed and preparing for a not yet known set of FR:s ahead of the life cycle start and stabilize through the application of configuration management.

In order to achieve the possibility for an ideal $\mathbf{DM}_m$ all integration has to be oriented towards the architecture. Furthermore the system should be redundant with N>M in areas where more functionality is expected, in order to be able to integrate future FR:s and maintain the ideal decomposition. This initial overcapacity in the design is an investment for response to future requirements and implies that the design solution is probably not optimized for its initial release, but rather optimized for providing the best value throughout its life.

The architecture is subject to FR:s that are essentially different from those that are included in one release of the system, those are e.g.

- provide possibility to include DP:s such that all expected FR:s for the systems intended scope over time could be met in a modular way.
- provide flexibility to expand over time
- provide high degree of invariance in the design matrix $\mathbf{DM}_a$
- support functional integrity over time
- simple interfaces
- support modular VVC

Each project to develop an initial release or upgrade of the system/aircraft where the architectural work have been successful is then subject to introductions and changes within **FR** and **DP** such that **DM** maintain its ideal properties with unchanged $\mathbf{DM}_a$ and Diagonal $\mathbf{DM}_m$ over time.

3.3 Architectural evolution

In equation 2b the ideal system design is established based on the ability to identify a $\mathbf{DM}_a$ that remains invariant and the evolution of the architectural performance and functional evolution are designed into $\mathbf{DP}_a$. It further prescribes the ability to maintain $\mathbf{DM}_m$ diagonal.

As long as the invariance of $\mathbf{DM}_a$ remains unchallenged there is little difficulty to maintain the system integrity and flexibility, but at a certain stage the effort to defend its invariance grows. This might be triggered from several reasons,e.g.:

- **FR** or **DP** are changing too much and the system expands beyond its boundaries of flexibility in terms of requirements or solutions. This issue should likely be addressed with the development of a new generation of the system.

- **FR** is diverging to a too wide coverage and the motivation for keeping the system together diminishes. The solution is to segment the system into a family of variants that share a subset of the system.

- The architecture becomes obsolete as the internal relations between DP:s and their importance is changing to a degree where the architectural design criteria are no longer valid

The architecture has a low clockspeed, or long life, which implies not only a stable behaviour in one generation of a system, but also that large fractions of the architecture is likely to be inherited between generations of systems and used in several variants depending on the segmentation structure.

The extent to which this is valid for families or generations of systems, determines the degree of commonality and hence to what level it is motivated to coordinate the supply chain and process between them. On the architectural level aerospace systems have many common FR:s, e.g. to support system integrity and airworthiness, and it is likely that the general process for this has a large commonality, while the sub- system content may differ to a large extent and motivate differentiated approaches to the development in the supply chain.

4 Conclusion

Aerospace and defense systems have been analyzed based on 3D concurrent engineering and axiomatic design as complex systems with long life cycles with particular attention to time and flexibility.

By dividing the design perspective into two domains, one long term architectural and one short term modular, and identifying an ideal product architecture for that situation, some PDC characteristics are identified in the three dimensions of product, process and supply chain. This time division enables to optimize the design in terms of integral architecture and modular flexibility and supports the resolution of the issue with conflicting short and long term goals.

Based on the ideal system architecture, some requirements on the PDC are derived. It follows that ideal conditions are heterogenous and all three dimensions should be designed and managed such that the PDC for various sub- systems can be adapted ideally to its conditions.

The flexibility searched in the product has to be reflected in process and supply chain, otherwise they likely will interlock each other. The design decomposition have to be reflected throughout the life cycle, otherwise the flexibility will be lost, e.g. verification and validation have to reflect the product decomposition.

In all, the possibility to develop and maintain an adaptible complex system with a long life cycle requires a holistic approach considering the three dimensions over time, with extensive attention and support for the long term perspective. Some enabling mechanisms have been proposed.

5 References

[1] Fine, C. *Clockspeed- Winning Industry Control in the Age of Temporary Advantage*, Perseus Books, 1998.

[2] Suh, N P. *The Principles of Design*, Oxford university Press, 1990.

[3] Suh, N P. *Axiomatic Design*, Oxford university Press, 2001.

[4] Wasner, R. *The Outsourcing Process – Strategic and Operational Realities*, Licentiate Thesis No. 163, Linkopings Universitet., 1999.

[5] Utterback J. *Mastering the Dynamics of Innovation*, Harward Business School Press, 1994.

[6] Fredriksson B; Holmberg G; Lilliecreutz J. *Collaborative Product Development Supported by Modelling and Simulation*, NATO AVT Modelling and Simulation Symposium, Paris, France, 2002.

[7] Oliver D; et. al. *Engineering Complex Systems with Models and Objects*, McGraw-Hill, 1997.

[8] Fredriksson B. *Holistic Systems Engineering in Product Development*, Military Aircraft, 1995.

[9] Holmberg, G. *Integrated Product Development - A key to Affordability*, ICAS 2000, Harrogate, UK, 2000.

[10] Frost R. *Simulation Based Acquisition –An Ongoing Outlook-* ,Simulation Interoperability Workshop, Fall 1998.

[11] Backlund, G. *Modeling Requirements in Systems Development - a Way to Manage Complexity*, Licentiate Thesis, Linkopings universitet, 2000.

[12] Kaplan R; Norton D. *The Strategy Focused Organization*, Harward Business School Press, 2001.

[13] Fredriksson B. *Designing for Best Life Cycle Value- the JAS 39 Gripen*, MIT/ LAI Conference on Enhancing Enterprise Value, March 29-30, 2000.

[14] Highsmith J. A. III. *Adaptive Software Development- A Collaborative Approach to Managing Complex Systems*, Dorset House Publishing, 2000.

[15] Murman E. M.; Walton, M; Rebentisch E. *Challenges in the Better Faster Cheaper Era of Aeronautical Design, Engineering and Manufacturing*, ICAS 2000, Harrogate, UK, 2000.