Processes and Beyond

Duarte Gonçalves
CSIR
PO Box 395
Pretoria, 0001
dgoncalv@csir.co.za
19 March 2008

Abstract. Systems engineering is about coordination. To coordinate is to bring the different elements of a complex activity or organization into a harmonious or effective relationship as a whole system. This paper views coordination in the context of unprecedented or innovative systems and attempts to focus some attention on important social issues relating to how work gets done. A framework of coordination mechanisms is developed. Coordination can be performed using processes, work products, skill/knowledge and mutual adjustment (personal or group communication). Under conditions of high task uncertainty, high task interdependence and dynamic environments, mutual adjustment is favored as opposed to process. Under these conditions this is not a choice. Most methodologies use some combination of the basic coordination mechanisms. The paper presents a case study illustrating some aspects of coordination.

Background

While contributing to a quality management initiative I became concerned about the amount of focus on process. Our organization is involved in research and development work where the systems are unprecedented or at least innovative in some sense. This is an important aspect of the business. In such cases, there are many uncertainties and knowledge application and generation are important outcomes. Within the organization there were some who argued that we should just move to agile methods. In some areas of the organization processes are used. It seemed that the question was: What mechanisms are available for coordinating engineering and *under what conditions* should we use a specific mechanism? What are the *fundamental* differences between process-based and agile methods? The intent here is not to fan the plan-based vs. agile debate (see for example (Boehm and Turner 2003) for a sober discussion) but to create a framework which explains the fundamental differences. Other concerns were that there had been previous quality attempts that had been unsuccessful. Simply imposing another quality system where there is a mismatch between the formalized management control system and the engineer's perceptions of the task will just lead to distrust (Sitkin and Stickel 1996). The issue was to more clearly define conditions for using a specific coordination mechanism.

Introduction

Systems engineering is about coordination. To coordinate is to bring the different elements of a complex activity or organization into a harmonious or effective relationship as a whole system[1]. The 'elements' could be sub-activities, technical knowledge, physical components, management and other aspects that would make up a system as a whole. The reference to 'harmonious or effective relationship' indicates the purpose of coordination. The system could be the development system, the production system or the operational system. Coordination could be across the life-cycle, or within one part of the life-cycle phase. The focus of this paper will be on the development system, i.e. coordination of work related to engineering the operational system. I will assume that the development team, which forms part of the development system, is co-located.

Because of size, complexity, specialization and multidisciplinary nature of projects, a division of labor arises. The division of labor requires coordination to achieve objectives. While many systems engineers place considerable focus on process, process alone is not always enough. This is not to say that we should abandon processes and embrace a new cure-all method – in fact processes are valuable tools. But the reality of everyday practice is that we use other mechanisms for coordinating but some of these mechanisms are hidden in anonymity. Plans and processes cannot cover every contingency when there are unknowns and there is certainly a relationship between coordination and risk.

In the management literature there has been extensive study of how coordination drives organizational structure (Mintzberg 1979). I draw on this literature and other literature to build a coordination framework in a systems engineering context. Factors driving the use of a specific coordination mechanism will be discussed. Finally, I present a case study to illustrate some of the ideas.

A Framework of Mechanisms for Coordination

The mechanisms of coordination are processes, plans, work products, skill and mutual adjustment (Mintzberg 1979) (refer to the framework in Figure 1). Coordination is based on agreement by those participating in the project. Agreement is shared knowledge in much the same way that natural language requires shared knowledge of words before we can communicate. In the case of coordination, the shared knowledge of agreement is a prerequisite for effective work. Team members who do not agree "should be rapidly identified and reassigned" according to (Boehm and Turner 2003). Standardization will be taken to mean agreement that persists over more than one project. In many cases the use of processes, plans, work products and skill for coordination leads to a reduction in coordination related communication *under normal circumstances* (Mintzberg 1979).

Processes are specified in terms of activities, sequences, rules, and tools to be used. Processes usually specify *what* needs to be done and, depending on how the processes are implemented within an organization, they can provide high levels of control over what is being done. One might start with a standard process or process reference model and instantiate it by means of a plan for a specific project with a schedule and intended resource allocation. By 'instantiate' I mean tailoring a process reference model for a specific use. A Systems Engineering (Management) Plan is an example of this. But plans do not have to be based on a standard or reference process. Processes and plans are impersonal and usually codified.

[1] Adapted for this use from the Concise Oxford Dictionary, Tenth edition.

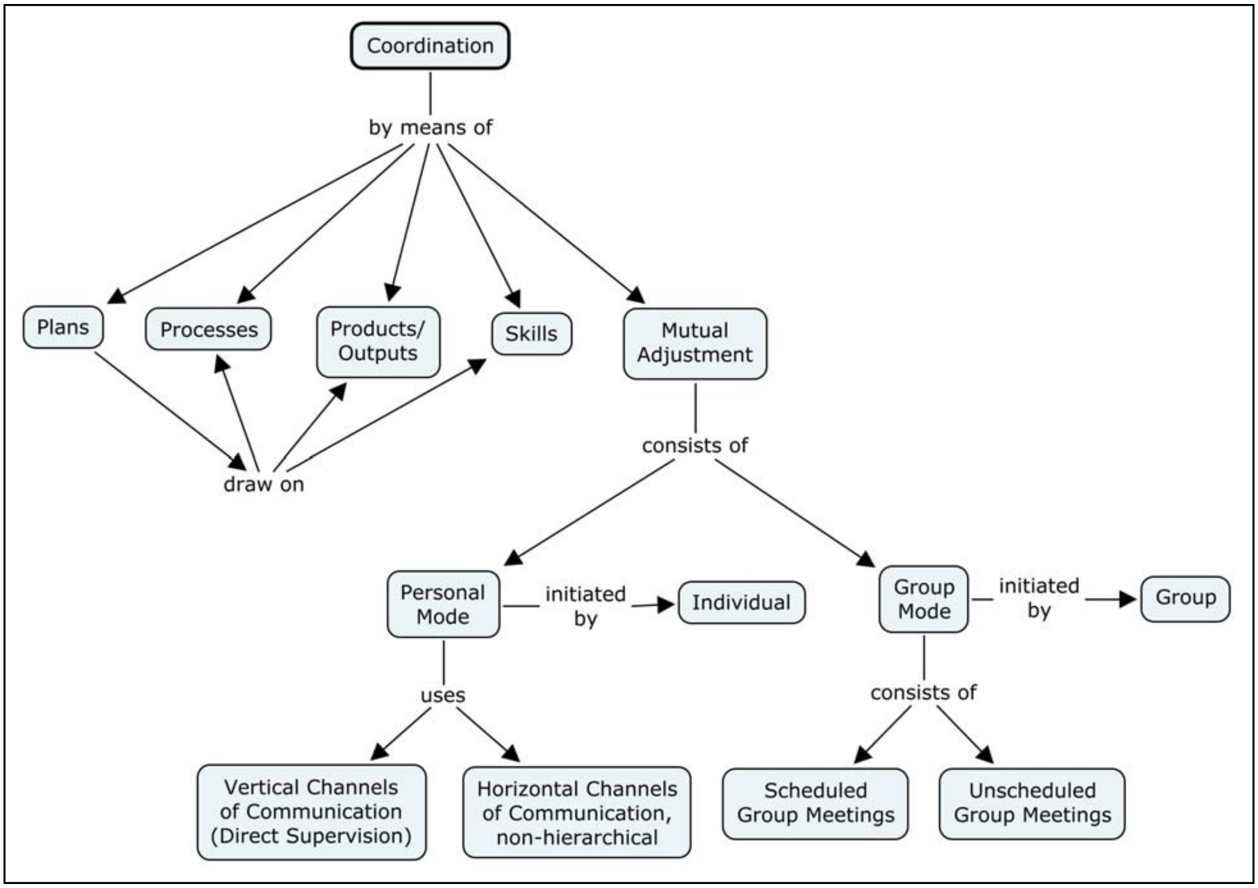

Figure 1 Framework of coordination mechanisms[2]

The use of work products or outputs as a coordination mechanism requires agreement on some characteristic or attribute of the product, but not necessarily the whole product. Work products as a coordination mechanism are useful in a number of situations. Agreement or standardization of work products creates a predetermined interface between tasks allowing work to continue independently on two or more tasks. An example could be simultaneous development of embedded software and its processing platform. Defining protocols, registers and data formats allows work to proceed concurrently on the software and hardware. But agreement or standardization can also be used to coordinate across life-cycle phases. Since some electronic subsystems can become obsolete in a matter of years, standardizing an aspect of the design like an interface or certain functionality, allows for replacing the subsystem with one that uses current devices. In other cases documents are standardized (for example MIL-STD-961D) in order to improve usability (like reducing time to find information) or completeness (attempting to reduce error by omission by providing a memory cue). If the document headings and numbering are agreed on, users who are familiar with the standard will know where to find information and authors will be prompted in terms of what information is required. In the area of modeling standardization of viewpoints, i.e. the specifications for constructing views, has emerged in the form of architecture frameworks (Maier and Rechtin 2002).

Standardization of skill and knowledge is a way of coordinating using the type and level of skill required to perform the task. If the process addresses the 'what', then the 'how' is largely a

[2] Based on Mintzberg 1979 with extensions to mutual adjustment from Van de Ven 1976.

matter of skill. The issue of ‘when’ to do ‘what’, i.e. the conditional knowledge is also critical. Skill takes over where process stops (Suchman 1990). There is a trade-off between skill level and process for various reasons. There may be a range of skill levels ranging from novice to expert. The higher the level of process abstraction the greater the level of skill and knowledge required. But, very detailed processes are not without problems because the more detailed the process becomes, the more sensitive or ‘brittle’ it becomes to context of application and unexpected events (Vincente 1999). Skill levels also cannot be defined independently of the organization and its business. An important point that arises is that processes should contain a specification of the required skill level for correct execution. Similarly, simply standardizing products can be dangerous unless an associated level of skill and possibly process are defined. In the context of Agile methodologies skill levels have been defined (Table 1). For systems engineering an important contribution in defining generic skill levels is the *Systems Engineering Competencies Framework* (INCOSE UK 2006).

Table 1 Levels of software skills (Boehm and Turner 2003)

Level	Characteristics
3	Able to revise a method, breaking its rules to fit an unprecedented new situation.
2	Able to tailor a method to fit a precedented new situation.
1A	With training, able to perform discretionary method steps such as sizing stories to fit increments, composing patterns, compound refactoring, or complex COTS integration. With experience, can become level 2.
1B	With training, able to perform procedural method steps such as coding a simple method, simple refactoring, following coding standards and configuration management procedures, or running tests. With experience, can become level 1A.
-1	May have technical skills, but unable or unwilling to collaborate or follow shared methods.

The simplest but also the most complex coordination mechanism is mutual adjustment. Mutual adjustment rests on communication. The communication could be initiated by an individual (personal mode) or a group when the project team consists of more than one group (group mode):

- Personal mode communication is either *horizontal*, i.e. at the same level in the organizational structure or *vertical*, usually involving a line manger, supervisor, or project manager and a subordinate (which Mintzberg calls ‘direct supervision’),
- Group mode communication in the form of either scheduled or unscheduled meetings.

Mutual adjustment relies on ‘know who’ before significant communication can begin (Johnson et al. 2002). ‘Know who’ is important for teams that are starting up, in particular, and also for communication across the organization. Proper scoping will identify authorities, responsibilities and roles (Hooks & Farry, 2000). These are also included in plans in the form of a roles and responsibilities matrix. These links allow team members to find the relevant person and start communication. But communication is not an end in itself. Communication forms the basis of coordination by providing feedback on the status of various tasks, problem-solving, accessing team member knowledge and building agreement on how to proceed.

Coordination depends on detecting issues or situations which may upset harmony or compromise effectiveness. Issues or situations requiring coordination may have been detected before the project, i.e. they are known. In such cases they may be captured in processes, requirements on products or simply as knowledge. ‘Lessons learned’ is an important source of these detected issues. But some issues or situations are only detected after the project starts. These are the domain of mutual adjustment. Reviews, for example a critical design review, are planned and conducted in order to detect issues. Others may be detected during verification or validation, or by accident. Once detected, additional information may be collected and plans, processes or other products revised. Once agreement is obtained, work continues.

The coordination mechanisms of Figure 1 are not mutually exclusive. Methodologies use a *combination* of these mechanisms and the exact combination determines the character of the methodology. Some evidence of this, in a software context, can be found in ‘typical days’ and ‘crisis days’ for plan-based and Extreme Programming in (Boehm and Turner 2003). A case study of a process/plan-based methodology will be presented later in the paper. Here we need to distinguish between process/plan-based methodologies of traditional systems engineering from processes and plans as coordination mechanisms. Each coordination mechanism has drivers or conditions under which it is useful.

Factors Driving Coordination

Having described the framework of coordination mechanisms, the drivers of process, plan, and mutual adjustment will be discussed in more detail. High task uncertainty drives mutual adjustment (with one exception) and a decrease in or suspension of process and plan-based approaches. There appears to be no correlation between task uncertainty and the use of personal mode vertical communication (direct supervision) (Van de Ven 1976), i.e. under conditions of high task uncertainty management is unlikely to be able to tell you what to do. Similar effects occur with task interdependence, but to a lesser extent. Factors driving *mutual adjustment* (personal mode, horizontal communication and group mode) are:

High task uncertainty: Task uncertainty (van de Ven and Delbecq 1974) refers to the difficulty and variability of the work. There are a number of indicators of task difficulty. Firstly, no clearly defined body of knowledge that can be accessed exists. This might be manifested in the unpredictability of solutions or actions taken. A second indicator of task difficulty is the time required to solve problems or tasks whose solutions are not immediately obvious. Thirdly, there is a delay between when a solution is implemented or action taken and the results are observed. This means that issues for coordination are delayed in their detection. The other measure of task uncertainty is the variability of work. One important measure of variability is the number of exceptions. Another measure is the variety of problems or conditions that make the work non-routine. High levels of task uncertainty may eventually result in re-planning, role changes, adjustments to schedules or priorities.

High task interdependence: Task interdependence is a measure of workflow interaction between team members in performing their individual jobs (van de Ven et. al 1976). More interaction is required, for example, when the knowledge required to complete a task is distributed across a number of personnel. On the one end of the interdependence scale is independent workflow, followed by sequential workflow and at the highest level of interdependence is team workflow, with a network of interconnections. Simply having a team working on a problem does not imply that there is team workflow since not all team members may be involved in the task.

More dynamic environments: Whereas task variability is largely a property of the task, the environment is external to the project team. Dynamic environments are the selling point for using Agile methods (Boehm and Turner 2003). As an environment becomes more dynamic, more effort is involved in re-planning, favoring mutual adjustment (Mintzberg 1979). The principle that "for continued system cohesion, the mean rate of system adaptation must equal or exceed the mean rate of change of the environment" can be applied to the developing system (Hitchins 2003, p109). A stable environment alone does not necessarily drive processes.

But mutual adjustment is prone to 'information loss' (Van de Ven 1976). Mutual adjustment and Agile methods rely more on 'tacit'[3] knowledge when less documentation is used (Cockburn 2002). While this enables faster response it also has a number of pitfalls. Firstly, it is difficult to compare 'mental models' for alignment and consistency. Secondly, when team members leave or the software team is disbanded at the end of the project, this knowledge is dispersed. Thus mutual adjustment may need to be balanced by the use of plans, processes and products for coordination. Factors driving *plans and processes* are:

Large teams: As Mintzberg (1979) points out, mutual adjustment is far from efficient exactly because of all the communication. This apparent lack of efficiency is in terms of both cost and time. For a work-unit with N people, there are N(N-1)/2 possible communications channels. Increasing team size drives the use of processes and plans (Boehm and Turner 2003 and van de Ven et. al 1976). Also, increasing team size leads to a decrease in the number of *unscheduled* group meetings because team members become more dispersed.

Criticality and the burden of proof: Two familiar but important drivers of process are criticality and what has been termed "burden of proof". Criticality refers to consequences arising from loss of life or loss of the system (Boehm and Turner 2003). Burden of proof, usually related to criticality, relates to providing a legally convincing argument of compliance. An example is satisfying the nuclear regulator that the reactor design is safe.

A project may typically consist of a number of tasks. Not all tasks will have the same task uncertainty, interdependence or sensitivity to environmental dynamics. This profile may vary over life-cycle phases and technologies requiring a customized blend of coordination mechanisms. Exploration, mitigation of certain risks and experimentation may require more mutual adjustment because of higher task uncertainty. For example, unpredictable external events outside the project's control may be better suited to mutual adjustment. As uncertainty is reduced over the life-cycle, process and plans becomes a more attractive *choice* in terms of efficiency. In the end the balance of coordination mechanisms is evaluated in terms of effectiveness which may include cost, time or robustness.

A case study is useful to make some of these ideas clearer and to illustrate how a combination of coordination mechanisms is used in practice.

Case study

An engineering project example shows how coordination changes from a largely plan and process driven approach to mutual adjustment on discovering an unexpected problem during validation. The project was of low complexity, but will serve to illustrate some of the coordination mechanisms.

[3] 'Tacit' is used in the same sense as in (Cockburn 2002).

Background

Five years ago, I led a team that designed and built an instrument for measuring dust density. At the time there were no such instruments available commercially for this application. The system consisted of a laptop with data-logging software and an instrument which attached to vehicle tow bar. The data-logging software was developed by our customer. The project team for the instrument consisted of a total of 6 people which covered the areas of optics, signal processing, electronics design and embedded software, mechanical design and systems engineering. Members of the team had worked together for at least two years prior to this project.

Plans and processes and products

The project (of building the instrument) was run largely using a plan developed at the beginning of the project. Each team member was assigned tasks during the planning. The tasks were defined around process steps: requirements analysis, design (architecture and detail design were not split in this case), and implementation tasks related to mechanics, electronics and embedded software. Verification and validation, also part of the plan, will be discussed.

The customer had decided that he would develop the laptop software. A simple protocol was negotiated with the customer and documented. This allowed the customer's team to develop the laptop software while our team was working on the optical instrument. This interface also allowed our team to verify operation of the instrument. Within the instrument team, several scheduled group meetings were held to discuss internal interfaces between internal mechanical, optical and electronic assemblies.

The verification strategy was in two parts. Firstly, signal processing concepts and analogue electronics were verified using simulation. This identified a major problem with temperature stability of an infra-red source. This discovery led to the inclusion of source temperature stabilization. Secondly, after manufacturing, a staged verification and integration approach was followed. The optical instrument passed various performance tests relating to stray light rejection, noise and linearity. Thermal testing confirmed the operation of the temperature stabilization loop. Since we had no access to vibration test facilities of the size required for the instrument, we could not test vibration. This was identified as a small risk. A final validation exercise was planned in the form of a field trail.

Validation and the discovery of the dust adhesion problem

The team had successfully designed other equipment for operation in dusty environments and some dust adhesion on the optical windows had been anticipated early in the design. The difference in this case was that this was an instrument for *measuring dust density* - a specific area which the team had not experienced. During validation, the customer and I discovered that the measured dust levels increased with increasing dust levels but remained high even when airborne dust had settled. Cleaning the windows of dust confirmed the problem. Vibration did not seem to be a problem, but we had discovered a serious dust adhesion problem. This was not in the plan. The customer was unhappy and called a meeting with my manager. Because the situation was sensitive and there was a potential for other orders, the line manager (who was not part of the team) allocated additional funding to investigate the problem and find a solution. I approached certain team members that I knew would be able to help and collected their ideas. I searched the Internet for clues. A total of ten possible solutions were developed over a period in three different categories: forcing dust off or away from the window, keep dust from the optical window and reducing dust adhesion.

Based on some informal discussions, one of the potential solutions, based on forcing dust off the window, was allocated to the mechanical designer to build and evaluate it. Poor performance and concerns regarding safety forced the team to abandoning this solution. During an informal visit to one of my colleagues and team member we came up with a simple and elegant solution, based on reducing dust adhesion. It did not introduce any moving parts, or any new problems. I investigated this further. Measurements indicated a 60% reduction in dust adhesion. A new plan was defined to design and modify the existing instrument, re-perform limited verification, and additional validation. Field tests indicated a dramatic improvement in performance. While dust was still adhering to the window, the reduction in dynamic range of the instrument was small. The problem had been solved.

Discussion

Since much of the solution could be modeled, there was low task uncertainty. A plan and processes were well suited to this phase since rework to the instrument hardware would just add cost. In this case study coordination on aspects of the product, namely the interfaces, was accompanied by mutual adjustment.

However the team could not predict the effects of the dust on the instrument. In this aspect the team faced some task uncertainty. When the problem was discovered the only option was mutual adjustment (although it was not stated as such), since the existing plan did not deal with this contingency. Once a solution was found the project continued on a plan.

Conclusions and Future Work

I defined coordination as bringing the different elements of a complex activity or organization into a harmonious or effective relationship as a whole system. While systems engineers place considerable emphasis on process, process is only one coordination mechanism and it can only deal with issues that are known. In systems that are unprecedented or at least have some innovative aspect there are issues that emerge after the project starts which require changes to plans, schedules and priorities. One could argue that this is the domain of risk management. There is certainly considerable overlap: firstly only detected risks can be managed (even then, one cannot be sure that they will be managed) and secondly risk management also requires coordination.

The coordination framework (presented in Figure 1) and factors driving the coordination mechanisms have value in identifying and assessing different coordination mechanisms for use on a project. Conditions of high task uncertainty, high task interdependence and dynamic environments may lead to specific issues and situations which can only be detected after the project starts. Under such conditions, mutual adjustment is not a choice but rather it is a phenomenon of social work under these conditions. Quality and process improvement programs should recognize the conditions under which the various coordination mechanisms can be used.

Instead of developing yet another methodology, it is worth investigating the building blocks of the current methodologies. This paper has presented a qualitative framework of coordination mechanisms. Future research into quantitative aspects of coordination would be useful. Incorporating activity and project success as well as parameters of the coordination mechanism such as the planning horizon amongst others would broaden this work. There is value to systems engineering in better understanding the basic coordination mechanisms and when each can best be applied.

Acknowledgements

I would like to thank Gavan Lintern and Alwyn Smit for their patience in discussing and reviewing this paper. I thank the INCOSE reviewers for their comments on this paper.

References

Boehm, B. and Turner, R. Balancing Agility and Discipline: A Guide for the Perplexed. Addison-Wesley Professional, 2003.

Cockburn, A. Agile Software Development. Addison-Wesley, 2002.

Hitchins, D. K., Advanced systems thinking, engineering and management, Artech House, 2003.

Hooks, I. & Farry, K., Customer Centered Products: Creating Successful Products Through Smart Requirements Management, American Management Association, 2000.

INCOSE, U.K., Systems Engineering Core Competencies Framework, 2006. Available from http://www. incose. org. uk/Downloads. Last accessed 06 March 2008.

Johnson, B., Lorenz, E. and Lundvall, B., "Why all this fuss about codified and tacit knowledge?", Industrial and Corporate Change. **11**(2), pp 245—262, 2002.

Maier, M. and Rechtin, E., The Art of Systems Architecting, CRC Press, 2000.

MIL-STD-961D Department Of Defense Standard Practice For Defense Specifications. Department Of Defense, 1995.

Mintzberg, H. The Structuring of Organizations. A Synthesis of the Research. Prentice-Hall, 1979.

Sitkin, S. and Stickel, D. Trust in Organizations: Frontiers of theory and research. R.M. Kramer & T.R.Tyler, eds., Sage Publications, pp196-215, 1996.

Suchman, L., Plans and situated action. Cambridge Univ. Press, 1990.

Van de Ven, A. and Delbecq, A. "A task contingent model of work-unit structure", Administrative Science Quarterly **19**(2) pp183-197, 1974.

Van de Ven, A., Delbecq, A. and Koenig Jr, R. "Determinants of Coordination Modes within Organizations", American Sociological Review **41**(2) pp322-338, 1976.

Vicente, K. J. Cognitive work analysis: Toward safe, productive, and healthy computer-based work. Lawrence Erlbaum, 1999.

Biography

Duarte Gonçalves holds a B. Eng in Electronics and a M. Eng in Computer Engineering and is currently employed by the CSIR. He has been involved in engineering surveillance systems for the South African DoD where he has extensive experience in electro-optical systems, ranging from modeling the environment and electro-optical observation systems, to signal and image processing. He holds a full patent in the area of imaging spectrometers. More recently, he has consulted to the Karoo Array Telescope project, the South African technology demonstrator for the Square Kilometer Array (SKA) as a systems engineer. In 2005 he won the best paper prize at the INCOSE SA conference. Mr. Gonçalves is involved in systems engineering coaching and quality activities within the CSIR.