Knowledge Integration – Balancing Between Anarchy and Despotism

Lars Taxén and Joakim Lilliesköld

Linköping University Post Print

N.B.: When citing this work, cite the original article.

Original Publication:

Postprint available at: Linköping University Electronic Press
http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-111352
KNOWLEDGE INTEGRATION – BALANCING BETWEEN ANARCHY AND DESPOTISM

Joakim Lilliesköld, Dr.  
The Royal Institute of Technology  
Department of Industrial Information and Control Systems  
Stockholm, Sweden  
Tel: +46 8 790 6869  
Fax: +46 8 790 6839  
E-mail: joakiml@ics.kth.se

Lars Taxén, Associated Prof.  
Campus Norrköping, Linköping University  
Department of Science and Technology  
Norrköping, Sweden  
Tel: +46-073-0977864  
Fax: -  
E-mail: lars.taxen@telia.com

Corresponding Author:

Dr Joakim Lilliesköld  
The Royal Institute of Technology  
Department of Industrial Information and Control Systems  
Stockholm, Sweden  
Tel: +46 8 790 6869  
Fax: +46 8 790 6839  
E-mail: joakiml@ics.kth.se
Abstract
A central issue in knowledge integration is how to achieve and maintain an optimal trade-off between differentiation and commonalization of knowledge. Too much emphasis on either side aggravates the efficiency of integration. The purpose of this paper is to make an inquiry into the trade-off problem from a particular theoretical perspective called the Activity Domain Theory. Two constructs from this theory are employed: the activity domain and the activity modalities. The activity domain frames the social fabric around actors working towards a common target, and the activity modalities are suggested as main dimensions through which humans coordinate their actions. We will utilize these constructs in analyzing a case study from ABB, a leading supplier of high-voltage equipment all over the world. Our conclusions are two-fold. First, complete commonality enforcement clashes inevitably with activity domain internal ideology. Second, management means such as images, need to be aligned with the activity modalities in order to achieve an optimal balance between anarchy, i.e., no commonality at all, and despotism, i.e., complete commonality. Based on these results, we suggest that the Activity Domain Theory may open up previously untrodden paths for theorizing about knowledge integration.
INTRODUCTION

The specialization and division of labour is characteristic for human activity. An inevitable consequence of specialization is coordination; the actions of individuals need to be synchronized in order to achieve common goals. Another, equally inevitable consequence is knowledge distribution; knowledge and capabilities are allocated to different individuals or groups. These basic facts remain the same, regardless of the specific modes of production; whether in a small group of hunters at the dawn of mankind, or in a modern enterprise spanning several organizations in the emerging global economy.

However, with increased physical dispersion, international separation, contingent cooperative ventures, mergers and acquisitions, etc., the complexities and uncertainties of coordination have increased. Moreover, technological advances have made it possible to develop increasingly complex products composed of parts from a multitude of technological domains such as software, hardware, mechanics, optics, etc. These trends have spurred a growing awareness of the importance of knowledge integration (KI), that is, how knowledge from various disciplines and locations can be engaged to reach a common goal.

The particular focus of this paper is the tension between differentiation and commonalization of knowledge. The efficiency of KI is dependent on achieving and maintaining an optimal trade-off between these two attraction points:

[Knowledge integration] depends upon the extent of commonality in […] specialized knowledge. There is something of a paradox in this. The benefit of knowledge integration is in meshing the different specialized knowledge of individuals - if two people have identical knowledge there is no gain from integration - yet, if the individuals have entirely separate knowledge bases, then integration cannot occur beyond the most primitive level. (Grant, 1996:116)

A key challenge in addressing the trade-off problem, as well as other issues related to KI, is to formulate a theoretical perspective that can be used for analytical / explorative and constructive / normative purposes. It is necessary to base claims on a solid, theoretical grounding of the concepts of “knowledge” and “integration”. Unfortunately, this seems not to be the case. From our point of view, several problematic issues can be identified:

• The very concept of KI is vaguely defined if defined at all. For example, Lehtonen et al. (2008) have identified five different perspectives from which KI is investigated. This leads to a lack of consensus about what the Unit of Analysis (UoA) in the KI discourse should be. As a consequence, knowledge accumulation about KI is aggravated.

• Various stances on the nature and acquisition of knowledge are more or less taken for granted and used without discrimination, for example: “knowledge may remain local” (Lehtonen et al. (2008); “capture and share knowledge across projects” (ibid.); “storing and accumulating of […] knowledge” (Cacciatori et al., 2008); “organizational remembering” (Cacciatori et al., 2008b); “artefacts for enhancing knowledge storage and retrieval” (ibid.). What does it mean for knowledge to remain ‘local’? What exactly happens when knowledge is ‘captured’, ‘accumulated’ and ‘stored’? Is it canned and sealed it in a bottle or put in a box on a shelf? How does an organization ‘remember’ something? Who or what does the remembering? How does an artefact for enhancing ‘knowledge storage’ differ from any other artefact? And so on. Without a clearly defined foundation for these and similar statements, arguments are inevitably weakened.

• Epistemological discussions are centred on the individual: “[If] knowledge resides in specialized form among individual organizational members, then the essence of organizational capability is the integration of individuals' specialized knowledge.” (Grant, 1996b: 375). Thus, the dialectic between the individual and her social embedding is de-emphasized.

• There is a direct, transparent link between the individual and the organizational levels: “the fundamental task of the organization is to coordinate the efforts of many specialists. (Grant, 1996:113). Nothing shields the individual in her every-day doings from the events taking
place at the organizational boundary. The heterogeneous structure of real organizations appears to be blanked out.

- The issue of knowledge integration between firms appears to be under-researched. Knowledge integration within-firms and cross-firms are treated differently.

In response to these issues this paper proposes the Activity Domain Theory (ADT) as a theoretical ground for making inquiries into KI. In this theory, the activity domain is the central construct. The activity domain frames the social fabric around actors working on a shared work object in order to achieve a common goal. The work object and the motive for its existence determine what resources are needed to achieve the outcome. A resource in ADT is the combined capability of actors and means utilized in purposeful actions. By engaging in the activity domain, a common worldview or ideology is enacted among actors concerning which phenomena are relevant and which actions are meaningful in the domain.

In the activity domain, two types of resources are used. Transformative resources are used to modify the work object into an outcome, and coordinative resources are used to coordinate the transformative resources. A frequently cited definition of coordination is given by Malone & Crowston: “Coordination is managing dependencies between activities.” (Malone & Crowston, 1994: 90). Since activities take place in a certain order, this definition implies a temporal dimension of coordination. Based on industrial experiences of coordinating complex systems in the telecom industry, Taxén has suggested that managing dependencies in the temporal dimension is necessary but far from sufficient (Taxén, 2003; 2004; 2005). A number of additional dimensions need to be considered: contextualization, spatialization, stabilization and transition. The ensemble of these dimensions is called activity modalities in ADT.

The choice of the activity domain as the central organizational construct implies that the domain is positioned as an intermediate construct between the individual and the organization. Individual knowledge and individual actions can only be understood in the context of the activity domain. This in turn implies that “knowledge” and “integration” should be investigated from the activity domain perspective rather than from the individual or the organizational perspective.

The activity domain is a generic construct; it can be equally applied to any constellation of work contexts: two persons working on the same task, a team, a functional unit in an organization, the organization itself, or cooperating organizations. All of these work settings can be apprehended as activity domains. Thus, the firm is conceptualized as a constellation of collaborating activity domains. As a consequence, the trade-off problem can be reconceptualised as a balancing act between two extremes that we label anarchy and despotism. Anarchy implies that there is no commonality among activity domains, while despotism indicates complete commonality. This enables us to paraphrase Grant’s paradox as cited above as follows:

[Knowledge integration] depends upon the extent of commonality in [...] specialized activity domains. There is something of a paradox in this. The benefit of knowledge integration is in meshing the different specialized activity domain - if two domains are identical there is no gain from integration - yet, if the domains are entirely separate, then integration cannot occur beyond the most primitive level.

The paper is outlined as follows. First we sketch the contours of the ADT. Next, we give an account of a case study performed at ABB, a leading supplier of high-voltage equipments all over the world. In their attempt to become the leading global system supplier, ABB performed a series of development programs testing different approaches. The case study, called Release 2.0 was the third development project with a similar scope. The prior two, Release 1.0 and Release 1.5 both failed in terms of time, cost and delivered functionality. Release 2.0 was however considered a success. These projects were positioned differently on the anarchy – despotism scale. In the first two projects, ABB used an approach close to despotism, meaning that a high degree of commonality was enforced across a large number of sub-projects (activity domains in our terminology) included in the overall project. As a consequence of the prior failures, Release 2.0 utilized an approach closer to anarchy. In a number of ways the previously enforced commonality was retracted, and more autonomy was given to the sub-projects. Simultaneously, power-
ful commonality means, among others a so called dependency diagram, were enforced for managing the coordination of deliveries from the sub-projects.

In the discussion that follows we interpret the case study with the ADT as a guiding framework. Our findings indicate that enforcing an intrusive commonality conflicts with the internal ideologies of the activity domains. Debates and disagreements about the ‘proper way of doing’ tend to slow down and even inhibit knowledge integration.

Means for enforcing the necessary commonality, such as the dependency diagram, need to be easily understood. Moreover, the dependency diagram exhibits several activity modalities as well as interdependencies between these. One-dimensional diagrams showing the temporal dimension only (such as Gantt schemas) are not sufficient. In conclusion, we suggest that the activity domain and the activity modalities provide guiding instruments for KI since they can be utilized for obtaining an optimal balance between differentiation and commonalization. Thus, the Activity Domain Theory may open up previously untrdden theoretical and practical paths for knowledge integration.

**THE ACTIVITY DOMAIN THEORY**

ADT originated in the development practice of Ericsson, a major supplier of telecom systems worldwide (Taxén, 2003). At Ericsson, ADT influenced and was influenced by efforts to coordinate large, extraordinary complex system development projects. The focus of ADT is the construction of communal meaning of how coordination shall be understood, something which turned out to be a major issue in the Ericsson practice. From the outset, an ambition with ADT has been to provide an operationalizable theoretical foundation that can be efficiently applied to demanding coordination tasks.

**The activity domain**

The central element in ADT is the activity domain. The activity domain is an off-spring from a branch of investigations that takes some kind of practice construct as the basic Unit of Analysis. The notion of practice has become influential in recent years in the organizational discourse (e.g. Schatzki et al., 2001). A practice is defined as “embodied, materially mediated arrays of human activity centrally organized around shared practical understanding” (Schatzki, 2001: 2).

Some examples of practice constructs are ‘social worlds’ (Strauss, 1985); ‘communities of practice’ (Lave & Wegner, 1991); ‘workpractice’ (Goldkuhl & Röstlinger, 2002); ‘work systems’ (Alter, 2006). However, the activity domain differs from these constructs in several ways. The philosophical roots of ADT are found in the notion of praxis (Kosík, 1976; Israel, 1979) and the Russian theory of Activity (Bedny & Meister, 1997; Engeström, 1999; Kaptelinin & Nardi, 2006). The activity domain builds upon and elaborates the ‘activity’ construct in activity theory. ‘Activity’, which is quite different from the ordinary English understanding of this term, was first introduced by Leont’ev as a fundamental unit in his investigations of the early manifestations of the mind in the human evolutionary history. Activities cannot exist without objects: “Any activity of an organism is directed at a certain object; an ‘objectless’ activity is impossible” (Leont’ev, 1981, in Kaptelinin & Nardi, 2006:55).

Moreover, activity is an inherent social construct in the sense that individual actions can only be understood in relation to the activity in which these actions are carried out. This was illustrated by Leont’ev by the famous example of the activity of hunting. Consider the roles of the beaters in this activity. Although the obvious motive for the activity is to get food, the actions of the beaters drive the quarry away. Taken out of the context of the activity, these actions appear to be meaningless if not downright misguided.

In the organizational context, an activity domain may be any organizational unit that produces some output that the firm needs in order to fulfil its goals. More so, even the firm itself can be regarded as an activity domain. In principle, any group of units within the firm, and of cooperating firms, can be apprehended as a constellation of activity domains. Thus, in ADT the activity domain is regarded as the kernel construct in organizational contexts; as the ‘DNA of the firm’, so to speak.
Knowledge acquisition
The debate on the nature of knowledge is an ever-lasting one. The literature on knowledge in organizations still “presents sharply contrasting and at times even contradictory views of knowledge.” (Brown & Duguid, 2001: 198). Traditional assumptions about knowledge “offer a compartmentalized and static approach to the subject.” (Blackler, 1995: 1021). It is common to see knowledge as “embodied, embedded, encultured and encoded” (ibid: 1021, italics in original).

As an alternative “knowledge (or, more appropriately, knowing) [may be] analyzed as an active process that is mediated, situated, provisional, pragmatic and contested.” (ibid: 1021, italics in original). A similar position has been advocated by Orlikowski (2002) who proposes to use ‘organizational knowing’ instead of ‘organizational knowledge’ to emphasize that knowing is enacted in practice:

Knowledgeability or knowing-in-practice is continually enacted through people’s everyday activity; it does not exist “out there” (incorporated in external objects, routines, or systems) or “in here” (inscribed in human brains, bodies, or communities). Rather, knowing is an ongoing social accomplishment, constituted and reconstituted in everyday practice. (Orlikowski, 2002, p. 252)

The enactment view of knowledge brings with it a connotation of ‘enactment towards what’? Enactment must be directed to something. Organizations are intentionally created to fulfill social needs. Consequently, knowledge in organizations is used for productive purposes:

Organizations are not basically knowledge systems, but systems that produce something of value “to the society. […] Only when the knowledge-creation process is set into the context of an organization’s activities, does the understanding of the knowledge processes help us understand organizational learning.” (Virkkunen & Kuutti, 2000: 297).

The primary role of the firm is in the “application of existing knowledge to the production of goods and services.” (Grant, 1996: 112). This position is also emphasized by Burstein & Linger (2003), who maintains that knowledge must be seen in relation to the task at hand. Instead of focusing on a philosophical discussion of the nature of knowledge, where the target of inquiry is ‘knowledge’ in general, the principal question to ask should be: “What kind of knowledge is needed in order to produce whatever the organization produces? What is the knowledge about?”

This view is foregrounded in activity theory by emphasizing the object of the activity – the work object – as the main constituent of the activity (e.g. Kaptelinin & Nardi, 2006). Work objects can be material or intangible things as long as they can be shared for manipulation and transformation by the actors. The existence of the activity domain is motivated by the social needs that the outcome fulfills. The work object and the motive are the key elements that distinguish different domains from each other.

The foregrounding of the work object of the activity domain means that individual knowledge is utilized in doing something to something; knowledge and means are enacted to make a change the work object into something that fulfills social needs. In order to capture the enactment view of knowledge, we suggest directing the attention to the more practice-related notion of capability. A decisive argument for concentrating on ‘capability’ is that capability can be equally applied to both humans and non-human things. In doing so it is possible to steer clear of the entangled discussion of whether knowledge can or cannot be embedded in artefacts and treated as a commodity. Through enactment – a recurrent interaction with available means – human capabilities and capabilities of means are mobilized as resources for actions in the activity domain. A resource, then, is simply a capability of humans and / or means that is meaningful to use in an Activity Domain; something that advances the state of the work object.

Transformative and coordinative capabilities
In the activity domain, actors perform various actions directed to the shared work object. It follows that two fundamentally different kinds of resources are needed: resources for the transformation of the work object into an outcome, and resources for coordinating transformative actions.

In order to analytically distinguish these two kinds of resources, the activity domain may be
comprehended as being in two, dialectically interrelated activity modes: the *transformative* and the *coordinative* ones. These two modes can be seen as two activities, however, with the transformative one as the primary, since this mode is directly related to the work object and the motive of the domain. Consequently, coordination is seen as an activity in itself:

Thus, by entering into cooperative work relations, the participants must engage in activities that are, in a sense, extraneous to the activities that contribute directly to fashioning the product or service and meeting requirements. That is, compared with individual work, cooperative work implies an overhead cost in terms of labor, resources, time, etc. The obvious justification for incurring this overhead cost and thus the reason for the emergence of cooperative work formations is, of course, that workers could not accomplish the task in question if they were to do it individually (Schmidt, 1990, in Schmidt & Bannon, 1992:8).

An example may clarify the relation between the transformative and coordinative modes. Let’s consider a requirement on a car. The content of such a requirement may be: “The car shall consume less than 0.5 litres per 10 km at a cruising speed of 100 km per hour”. The work object – the car – must fulfil this requirement. In addition there might be a multitude of other requirements on form, safety, exhaust limits, and the like. In order to coordinate the actions in the transformative mode, there is a need to keep track of all the requirements, preferably in a requirement management tool. To achieve this, certain coordinative capabilities must be enacted the activity domain, such as unique identifiers for each requirement; a set of states indicating what state a requirement is in (for example, whether the requirement is fulfilled or not); attributes characterizing the requirement; relationships to other items such as what specific customer has issued the requirement, and so on. In the coordinative mode only coordinative aspects of requirements are relevant, not the actual content of the requirement. The content, on the other hand is relevant in the transformative mode but not in the coordinative mode.

The outcomes in the coordinative mode are information models, process models, rules, IS support, etc.; items that are relevant for coordinating actions. In the transitive mode, these coordinative capabilities are utilized in the transformation of the work object into an output. Thus, the activity domain unfolds through an ongoing focal shift between the coordinative and transformative modes, in which transformative and coordinative capabilities are enacted. This is illustrated in Figure 1:

![Figure 1: The two modes of the activity domain](image)

**Coordination between activity domains**

Since activity domains differ with respect to their work objects and motives, meaningful transformative and coordinative actions in one domain may be meaningless actions in other domains. Each domain enacts its own particular worldview of what actions are relevant and useful. Even if two domains have the same kind of work objects and motives, like two automobile workshops servicing the same make of car, they would still enact domain specific capabilities. This is the main reason why it is necessary to introduce a practice construct like the activity domain as an intermediate construct between the individual and the organization. The determinant of individual capabilities and meaningful actions is the activity domain, not the organization. In fact, the only case when an intermediate realm is not needed is when there is no division of labor, that is,
when there is but one actor in the organization. This would be the case for a one-man activity such as a painter, a sculptor, etc. However, even in such cases, the activity domain has to interact with other domains, for example, a painter buying her equipments from a store.

Therefore, it is necessary to take into account coordination between activity domains in addition to coordination within domains. Consider the example with the car: One activity domain may be providing the motor of the car, while another provides the chassis. It is evident that the chassis has to be in place before the motor can be attached to the chassis. This dependency between activity domains needs to be managed as well as dependencies between actions within each domain.

These considerations indicate a key tenet in ADT: The coordination of actions within an activity domain is in principle not different from the coordination between activity domains. The coordination between domains takes place in an overarching domain with its own work object and motive. This overarching domain may be the firm itself. Thus, the transformative and coordinative modes are found in any activity domain, regardless of the particular position of the domain in a constellation of domains.

In Figure 2, this idea is exemplified for a telecom system provider such as Ericsson. The Ericsson company is considered as an activity domain that provides 3G systems to some customer. In order to do so, a number of activity domains are mobilized: Market & Sales working with customers and tenders, Research & Development working with the product, Supply & Implementation working with orders, and In Service Support working with the installed base of products. Each of these domains enacts different transformative and coordinative resources, and each domain can in turn employ the capabilities of other domains. Some of these may be common for several domains, such as a domain providing IT services to the entire organization.

Figure 2: The organization as a constellation of activity domains

The Activity Modalities
We have stated that the activity domain frames meaningful actions with respect to the work object and motive of the domain. A premise for meaning construction is the biological “substrate” inherited during the phylogenetic evolution of mankind. Even if humanity shows a tremendous linguistic variety, we can expect that the construction of the meaningful exhibits common patterns, simply because we are all human beings. Thus, we may hypothesize that meaning construction is in some sense trans-situational. In other words, regardless of the particular motive and work object of an activity domain, certain regular features should prevail between different domains. An indication of such features is provided by the empirical observations from the Ericsson development practice (Taxén, 2003). The analysis of these observations indicates that the evolution of activity domains proceeds along certain, dialectically interdependent dimensions, which have are coined activity modalities in ADT.

For example, one activity modality is temporalization. The enactment of capabilities in this modality results in meaningful artefacts such as clocks or calendars indicating a temporal dimension. Examples from the organizational context are business process models and Gantt diagrams. To become effective in the organization, the actors using such models must acquire
In the analysis of the empirical results from Ericsson, the following activity modalities were found to be particularly important for coordination (Taxén, 2003):

- **Contextualization:** Denotes the human capability to focus our attention. Our visual system simplifies a visual scene into a figure attended in the foreground and other things unattended in the background (Jackendoff, 1983: 42). Thus, a capability to contextualize appears to be innate in humans. Contextualization implies that the way we apprehend things depend on the context. For example, a grand piano is apprehended very differently in the concert hall and during transportation. Organizational manifestations of contextualization are, for example, organizational units, teams, and projects.

- **Spatialization:** Denotes the human capability of enacting spatial structures that signify which entities are relevant in activity domains, how these entities should be characterized and related to each other, and what state or condition they are. Examples of organizational manifestations of spatialization are information models, product structures, and conceptual models. Spatialization can refer both to direct, physical objects or signs referring to such objects. For example, both a map of a city and the spatial outline of the city itself are examples of spatialization.

- **Temporalization:** Denotes the human capability of enacting temporal structures that signify actions and the dependencies between them. Temporalization corresponds to the definition of coordination given by Malone & Crowston: “Coordination is managing dependencies between activities.” (Malone & Crowston, 1994: 90). Examples of temporal elements are business process models, interaction diagrams, and use cases.

- **Stabilization:** Denotes the human capability of enacting stabilizing elements that signify which actions are regarded as valid and useful in a domain. Stabilization manifests itself as norms, values, habits, routines, methods, rules, standards, domain specific languages, etc. Without stabilizing elements, coordination is impossible. They have the function of “... reducing the infinite number of things in the world, potential or actual — to a moderate number of well-defined varieties” (March & Simon, 1958: 181). Organizational manifestations of stabilization are, for example, naming conventions, business rules, and standards.

- **Transition:** Denotes the human capability of enacting elements enabling the transition between domains. Since communal meaning differs between domains, a particular outcome from one domain may be characterized differently in other domains. Transitional elements provide a mapping and translation between domain specific meanings that enables the actors to cooperate. Organizational manifestations of transitional elements are, for example, interface specifications and dictionaries for translating between organization specific languages such as product identification conventions. The transition modality enables the collaboration of activity domains.

In plain language, the modalities can be apprehended as follows:

- Human actions take place in social fabrics (contextualization)
- In order to act, humans need to orient themselves spatially (spatialization)
- Actions are carried out in a certain order (temporalization)
- Rules, norms, etc., signify valid and meaningful actions (stabilization)
- Humans cooperate across social fabrics (transition)

The activity modalities are dialectically interrelated, meaning that a change in one of them impacts all the others. They form a complete whole in coordinating human actions. If one of the modalities fails, coordination will be deprived.
It should be underlined that the modalities are grounded in the biological constitutions of humans, which indicates that modalities originate in the individual. The individual acts of writing a document, playing an instrument, painting a piece of art, and so on, are all coordinated according to the modalities. Thus, in social settings, the same modalities are at play; however, with the addition that individual actions need to be coordinated. In principle, there is no difference between my personal reading of a calendar and my reading of a common project plan in an organization. Both the calendar and the project plan are expressions of temporalization.

The enactment of capabilities along the activity modalities in the activity domain constitutes an ideology, that is, a wide-ranging system of belief or thought. Some elements of the ideology may be common to several domains, but in general, these vary between domains. In Figure 3, a conceptual model of the activity domain is depicted.

**Knowledge integration reconceptualised**

From the activity domain perspective the issue of knowledge integration is inextricably related to the activity domain. Knowledge is understood as the capability of actors to mobilize capabilities of means in order to become resources in the domain. Resources are always related to the motive and work object of the activity; meaning that capabilities that are useful in one domain may be useless in other domains. Moreover, even if two domains have identical work objects, the knowledge needed will still differ due to the historical evolution of the domain and the enactment of capabilities.

The recursive nature of the activity domain means that activity domains may employ other domains as resources. This is illustrated in Figure 4.
An activity domain (A) is employing four other activity domains (a1, a2, a3, and a4). Besides utilizing these domains for transformational purposes, there may be transformational resources that are specific for domain A. All transformational resources need to be coordinated by employing coordinative resources, again specific for domain A. This pattern is then repeated in each of the other domain, which in turn may employ other domains, and so on.

With this general pattern in mind, KI can be conceptualized as follows. Integration is understood as the coordination of transformative actions. Accordingly, knowledge integration is seen as the management of coordinative and transformative capabilities employed in activity domains. Since each activity domain enacts its own specific transactional and coordinative capabilities, KI becomes an issue of how far the coordinative capabilities of one domain shall be commonalised across other domains. Imagine, for example, that the activity domain A in Figure 4 is a consortium employing the firms a1 to a4 for a building project. How much should the coordinative capabilities of A such as article identification rules, project planning tools, routines, etc., be allowed to penetrate into the more or less independent firms, each of them having their own ways of doing things?

It should be obvious that KI is an issue of balancing between two extremes: no commonality penetration at all or complete commonality penetration. Since we want to phrase this issue in terms of activity domains, suitable metaphors for the endpoints may be ‘anarchy’ (“a state of lawlessness or political disorder due to the absence of governmental authority” and ‘despotism’ (“a system of government in which the ruler has unlimited power.”)1 Neither of these extremes is productive for KI. The optimal commonality penetration is a matter of balancing between anarchy and despotism (see Figure 5).

---

THE ABB CASE – Release 2.0

The studied project (Eriksson et al, 2002) was conducted within a group of companies; ABB that was created in 1988 by a merging of Swedish ASEA AB and Switzerland based BBC Brown Boveri. The group was affected by changes in the business climate, and since 1988 numerous acquisitions have been strategically made world wide creating a group with some 550 facilities (end of 1999). During 1999, the group accelerated its shift from delivery of standalone products to total system solutions.

Traditionally, sets of products had been developed within each company. The logistics had been fully refined around these products such as development, production, marketing, education, and sales. But the competition on the market made it necessary to be able to design, develop, engineer, and maintain systems easier and faster. The differences in the backgrounds of the companies presented difficulties when the standalone products were to be integrated into one system.

More implications were that problems occurred when products from different companies within the group were integrated into a system. This was due to the rapid development of product platform technology and products’ complexity. Much effort was invested in all major system deliveries in terms of the adaptation of interfaces among the different products in order to make them compatible. Further, the growing complexities of products and cross-functional integration of systems created a need for structured engineering and maintenance processes that could handle the growing complexity.

The studied project was one of several attempts to create a common product platform for substation automation systems within the power industry. The project was initiated by the group’s headquarters in Zürich, and should ensure that a set of products developed at each company could be easily integrated into system configurations determined by customer requirements on the global market. Prior to this project, two projects (Release 1.0 and Release 1.5) with similar goals and organizational structure had failed. Release 1.0 was an internal release, 1.5 had a customer.

Project Description

The aim, as well as the challenge of the project was to quickly develop a substation automation (SA) platform while retaining high quality. The nature of the products was such that lack of quality could easily cause damage to both personnel and material.

The project included the development of a new platform with a number of hardware and
software products, such as control and protection terminal units for medium and high voltage (HV), communication, intelligent gas insulated stations (GIS), substation automation monitoring system, engineering tools, etc. Also, a number of new functions were added to the group’s existing products as part of the project, such as new functions for a medium voltage feeder and motor protection, new functions in HV protections, support for intelligent GIS in HV products etc. One of the most important objectives of the project was to reduce engineering time and complexity.

Included in the project was the delivery of a pilot system to a customer in Australia, who should interconnect two power grids. The pilot system was essential to the customer, who required the system to be delivered on time. The customer followed the project progress closely during the project, but began to express doubts that it would be finished on time.

At an early stage of the project, it became clear that requirement engineering should be more synchronized. Product managers in each company were appointed the responsibility of defining market requirement specifications for standalone products as flexible components in a substation automation system for power transmission. However, time constraints from the customer forced an early start of the implementation in order to deliver the system functions to the end customer according to the requirements. Thus, instead of market requirement specification provided by product managers, functional design specification (FDS) created by R&D personnel were created for each product and/or function. A priority and importance value was assigned to each function. Functions of importance required to complete the product platform, but were not important to the customer, had low priority in the project. Since the FDSs were based on the functional list, technical characteristics were free to be interpreted by R&D personnel.

**Project Organization**

The overall project Release SA 2.0 consisted of several product development projects, with each project consisting of numerous of subprojects. The product development projects were dispersed according to the location of the core competence. When the activity was at its peak, about 200 people were involved in different projects that were located in one or more regions of Sweden, Switzerland, Finland, Germany, Italy and USA, see Figure 6.

![Project organisation SA 2.0](image)

**Figure 6:** The project organization

The overall project Release SA 2.0 was divided into three organizational levels; overall project management, project coordination at business area units, and subprojects. Within these three levels, six were important functions or roles:
The release manager
The release manager headed the project organization together with an overall project management team. He had the task of managing interfaces in different organizational and relevant functional levels from top management to product management, manufacturing, R&D management etc. as well as delivery to the end customer. Furthermore, his task included creating a project climate, removing obstacles, and coordinating managerial functions. Since the system specification group was included in the overall project management team, part of the release manager’s responsibility was also approving the system specification requirements and the validation of developed products.

The program manager
The program manager, was part of the overall project management team, and was an expert responsible for a particular knowledge area, e.g. Control Terminal Unit.

Steering Committee (STECO)
The Steering Committee of the project (STECO) included segment managers and other top managers of the group. Even though STECO did not meet that often, the project's progress was reported to them monthly. STECO could make general decisions, e.g., when the interests of the project and local companies were in conflict concerning the terms of global market strategy. For that purpose, Steering Committee meetings gave good support from top management to the project.

STECO also had the function of reminding involved local companies about the importance of keeping focus on global strategy. Although STECO was a very important element, especially at the beginning of the project when the interests of local companies could potentially jeopardize the project goal, it soon became obvious that a number of technical and more urgent issues had to be solved as well.

Configuration Change Board (CCB)
The Configuration Change Board (CCB) was formed in order to support the overall project. The purpose of the CCB was to decide on and manage changes at release level for the project. The CCB consisted of key persons, line, project, and product managers. The CCB was not a forum for general discussions. This meant that every issue on the agenda was prepared in advance so decisions could be made efficiently. Issue brought to this forum were: 1) change requests influencing the overall time schedule, 2) change requests concerning more than one product/sub-project, 3) when the functionality of the release or of a product were concerned. The group met physically once a month, but when required it could have a phone-meeting every week to make operative and urgent decisions in terms of, for example, development priority and functionality of single products. The CCB meetings assured that acceptance of information over unsolved issues was confirmed, decisions over problems were being made and that all members of CCB were committed to the new decisions.

Subproject coordinator
Subprojects were coordinated at a single geographic location by a project coordinator. Project coordinators were responsible for coordinating and resolving all issues within projects, and when necessary, initializing the decision process. Further, this function meant reporting the overall status of the coordinated projects to the release manager, and coordinating reporting at release meetings by deciding which subproject managers needed to participate.
Subproject manager
Each subproject team focused on a single product and had its own project manager. As the subprojects were autonomous, the subproject managers and personnel were approved by none or very little involvement from overall project management. The responsibilities for each project manager included defining functionality of the product under development and being responsible for the project plan. The subproject manager also had the responsibility of updating the time and milestone plans, and was responsible for the project follow-up and reporting. Further, the subproject manager had to request and keep track of deliverables from other subprojects.

Information management
Since the project was geographically dispersed, focus was on information management. At the beginning of the project, it was considered important that people meet and establish a personal relationship with each other. This was done in workshops early in the project when the overall project plan was developed. Of course it was impossible to gather all the project members together at the same time, but the priority was that at least all project managers and key personnel should meet. The main idea with this team building was that each team could later work as independently from each other as possible, but when needed, the different projects would know whom to contact in the other projects. The information management was handled with the following means:

Release meetings
When complex technology is being used and developed, it is natural that problems occur. It was considered important that all the project members were aware of the actual problems, so the release meetings served to address these important issues and identify the dependency between the different projects.

Release meetings were also used as a way to get all of the project members to accept the same goals and to co-ordinate interfaces among organizations, in other words to assure a common understanding of work moving towards common goals.

Further, these meetings were an excellent opportunity to make sure that all project teams and key persons were committed. Half the success has already been achieved if the common spirit is to assume that "everyone is doing his/her best". Otherwise, it is very easy to encounter a situation where one team blames the other.

Information Meetings
Information meetings are a way to inform project members at local organizations about the project’s overall status. These meetings were also used for less formal activities in order to motivate the local team.

Video and Phone Conferences
Videoconferences are considered a cost saving alternative compared to face-to-face meetings. The investment in compatible required equipment was not part of this project, but rather, a part of the group’s global strategy. The video conference system was considered very helpful to the project.

Project Database
A special database dedicated to the project was created. All relevant project information, such as functional design specifications, delivery plans, time schedules, notes from meetings, project related decisions, project diaries, project weekly reports, were stored in a database to which all project members were granted access to. The database was created in Lotus Notes and was accessible at every project location. This task was relatively easy as the common communication interface within ABB is Lotus Notes, and all project members were familiar with the program.
The project's database had several facets. The first was to ensure that every project member and line management knew the project goal, and that all the eventual changes were easily accessible and always available. The second was to have one place where all questions were collected. One of the categories in the database view was; “decision”, where proposals for problem solving could also be suggested. Whenever a question appeared in any of the subprojects where an agreement or decision was required, it was noted in the database. Concerned persons automatically received information about the notification and tried to come to an agreement usually via telephone or videoconference. If they could not agree, the first step was to contact the release manager and thereafter, the CCB.

E-mail
E-mail is used frequently and is considered the best form of quick information exchange. For information distribution though, it can have its disadvantages in comparison to using a database. This is because in a project of this size, project members are constantly changing, and it would take considerable administrative effort to collect all e-mail lists at each project member location. It could even create a hostile working atmosphere if someone is forgotten.

The Project Control System
The project management team needs to establish a system or control process to ensure they receive correct information about time, costs etc. In the project-planning phase, great effort was put in an attempt to reduce the R&D costs as well as following up the project budget according to the trends of R&D globalization within the business community (Gerybadze, 1999). This approach increased the amount of project management administration dramatically. It was almost impossible for the release manager to find out what was behind the reported figures. Therefore, the idea of project central budget control was abandoned, and budget responsibility was transferred to the local companies. The release manager was only responsible for the follow-up of costs. The focus of the project management was moved to a few, but qualitative measurable outcomes. Thus, what the overall project management focused on was what was accomplished at the end of the project.

At the first release meetings, a plan of dependencies for subproject deliveries was defined, the dependency diagram (see Figure 7). At this meeting it was also decided what should be developed and the budget for each development effort. All the involved organizations were involved with key personal at this meeting, and the result was that the project was well anchored in the organizations, allowing the organizations to use the work approach they were used to (i.e. follow their internal process and project handbooks).
This diagram served as a base for managing the overall project time schedule and controlling internal deliveries. The intention was to make the impact of subprojects’ delays on the common project schedule visible. The dependency diagram demonstrates the different product development activities, on the vertical axis to the left, as part of the system solution (which are the activities within the rectangle). There are 4 different projects here: two of them (SYS 500 and SMS 510) are different general product platforms that deliver something to the Power Link Project (the customer project in this case), for instance the user manual, or generalize the R&D result from this project. The last one, PVC is a project that will take part of the system solution included in the release and have them verified within a certain time after the delivery. The circles represent events in terms of releases, each one marked with the date of delivery and whether the delivery is part of an “alpha” or “beta” release. The arrows indicate which other releases the current one depends on and its influences.

Besides showing very clearly expected input and output of every subproject, the dependency diagram was also a good tool for defining common language, i.e. common understanding of the “alpha” and “beta” status of a product. The dependency diagram in Figure 7 has been modified, all the dates have been removed (there were dates for every release), and information such as when simulators were needed and which releases were common for all products has been removed from this version.\(^2\)

Another important step in simplifying the project control phase was the way project status was reported. Instead of trying to find a compromise on which common tools for project management should be used, it was decided to use Power Point slides with milestones, or decision

---

\(^2\) The dependency diagram in Figure 7 has been modified. All the dates have been removed (there were dates for every release), and information such as when simulators were needed and which releases were common for all products has been removed from the figure.
points. The key issues; milestones, and decision points, were highlighted in different colours in accordance with their status. The scale itself was green for issues running according to schedule, yellow for issues with high risk to be late, and red for issues running behind schedule, i.e. traffic light control.

**DISCUSSION**

All the involved organizations in the project were individual units that developed and sold stand-alone products. If a customer wanted to purchase a system solution, there was no developed framework for how to cooperate. And, since delivering a system solution was a growing trend, it was considered important to be able to cooperate. Further, this was also necessary in order to make use of all the competence the company had obtained in the process of mergers and acquisitions that led to the situation of having many stand alone companies. The challenges however, were not only to get the units to collaborate because of cultural problems, but that some of these units used to be competitors with competing technologies who now wanted to become ABB standard. In these attempts (Release 1.0, 1.5 and 2.0) different strategies were used.

With the Activity Domain Theory as a guiding framework the experiences provided by the ABB case can be interpreted as follows. The failed releases 1.0 and 1.5 were positioned too far towards the despotism side on the anarchy – despotism scale. Release 2.0 represents a move from the despotism end point towards a position closer to the anarchy side.

![Knowledge integration efficiency between the ABB releases.](image)

The movement towards a more balanced position was accomplished by the simultaneous withdrawal of intrusive commonality among sub-projects and enforced coordination commonality using mandatory coordinative means, above all; the dependency diagram. In the following, we examine these two movements in more detail.

**Commonality withdrawal**

We interpret the problems Releases 1.0 and 1.5 had as an indication of being too intrusive on Activity Domain specific ideologies. In this section, we list evidences from the ABB case of commonality withdrawal between Releases 1.5 and 2.0. Such evidences signify an attempt to move towards the anarchy end point.
Project model
The deadline and the content of the Release Project were set. The common milestones were also decided. But, the road to get there was more flexible. Every subproject was allowed to follow the project model and use the project tools and development tools they were used to. There were thus two decision structures; one given by every local R&D organization and one given by the overall project. The people that should be involved as well as sub-project managers were also appointed locally, without involvement from the Release project.

Budget responsibility
It is more difficult for top management to make sure that involved local organizations really work cost effectively with their project tasks if there are no formal contracts. In this project, the costs were managed on a local organizational level, which implies that it was almost impossible for overall project management to control them. Detailed control of costs on the overall project level required a big administration and a lot of time and energy; efforts which were instead spent on the achievement of the customer’s requirements and keeping focused on a time schedule and qualitative deliverables. The general opinion within project management was that flexibility in an R&D project of this size and scope contributed more to the project’s result than detailed definitions of responsibilities by means of formal contracts among involved companies.

Steering groups
There were two levels of steering group in the project. The Release Project steering group consisted of top managers from the involved companies as well as top level managers from the central organization. The purpose was to create commitment by stating that the project should be given priority, and by showing the interest from top management. This committee had the power to make decisions if the project and a local organization's interests were in conflict.

Then there were steering groups in all the local R&D projects, but they had a limited right to make decisions on the content of the project. They could only make decisions on issues that did not have a direct effect on any other subproject or the overall release project.

Commonality enforcement
The enforcement of commonality was done in several ways, such as the follow up of the project status by a common Power Point slide showing the status of decision points in traffic light colours. However, the main coordinating instruments were the dependency diagram and the CCB.

The Configuration Change Board
Early in the project, the Release project manager realized that there was a need for mutual interaction handling and solving of technical issues regarding the integration of the system components. The steering committee of the Release project was found to be ineffective in handling these issues. The participation on steering committee meetings turned out to be a very sensitive political issue. Therefore, the CCB was created, which could respond more rapidly to the control needs that occurred during the overall project. In the studied project, the creation of the CCB was a success. CCB could meet more frequently and had the mandate to act like the steering committee of the project for technical issues. By separating strategic decisions from technical ones, management support of the project becomes more effective; minimizing the delays of the internal project’s deliveries and misunderstandings between subprojects.

The coordination meetings
In order to create an understanding of interdependencies among subprojects in the initial project phases, frequent project meetings were used. Release meetings were coordinating meetings where all key persons from subproject manager on up in the hierarchy, met and coordinated these actions. These coordination meetings were held every 4 to 6 weeks. All other meetings
were ad-hoc when necessary; initiated either by a subproject having a problem that would affect another project, or by the CCB.

The dependency diagram

The dependency diagram allowed the involved organizations to work independently from each other. Further, this diagram provided a definition and understanding of dependencies of internal deliveries among subprojects. By means of the diagram, local project managers obtained a better understanding of the way other projects were dependent on the output of their project and how the overall project time schedule would be influenced by eventual delays. The diagram was simple to understand and updated regularly. It was a very powerful tool for anchoring the project’s goal in the local organizations and subprojects. This illustration, common for all subprojects, shows some interesting features from the activity modality point of view (see Figure 9):

![Diagram showing dependencies](image)

Figure 9: The dependency diagram seem from the activity modality perspective

The main feature of the diagram is to display the dependencies between three modalities (spatialization, temporalization, and transition). The spatial elements are lined up vertically (‘580 Platform*3,41’; ‘CAP 580*3,41; ‘REC 580*3,41’; etc.). The temporal elements are lined up horizontally (‘System definition’; ‘Test specification’; ‘Engineering of system environment’; etc.). The temporalization modality is manifested by circles, indicating a certain delivery to or from sub-projects (activity domains). In addition the end product, the delivery to the customer, is highlighted in the diagram. The interrelationships between the modalities are indicated by the arrows.

It is interesting to compare the dependency diagram with a traditional illustration; a process model from Ericsson (see Figure 10):
Most of the elements in the dependency diagram can also be seen in the process model: activity domains, spatial and temporal elements, and customer deliveries. The spatial elements are lined up vertically and the temporal ones horizontally. However, the interdependencies between these modalities are lost in a confusing jungle of dependencies between actions. The spatial elements, which are clearly seen to the left, vanish from the surface in the centre of the model, and resurface miraculously to the right.

Our interpretation of the dependency diagram is that it is congruent or aligned with the activity modalities. These are displayed in such a way that each modality is easily recognizable and separated. Moreover, dependencies between the modalities are easily identified. The process model, on the other hand, does not have a clean structure from the activity modality point of view and is much harder to make sense of. Consequently, this model is less useful as an instrument for action. It is interesting to note that the dependency diagram shows the same structure as another diagram, the so called Information Flow Diagrams (e.g. Taxén & Svensson, 2005) that were used at Ericsson as an alternative to the traditional, activity based process models such as the one in Figure 10.

To summarize, coordinative means such as images shall be structured in congruence with the activity modalities if they are to be effective as instruments for collective actions. This conclusion is reinforced by similar findings elsewhere (Taxén & Lilliesköld, 2008). Images shape the way we make sense of the world and contribute to our understanding of what actions are valid and meaningful in certain situations. For KI, the difference between an image aligned with the modalities (such as the dependency diagram) and images that are not aligned (such as the Ericsson process model) can make all the difference between a successful and failed project.

CONCLUSIONS

Achieving an optimal trade-off between differentiation and commonalization is a core concern in knowledge integration. We claim that current approaches dealing with this issue are based on inadequate theoretical perspectives. The theoretical framework proposed in this paper is an attempt to remedy this situation. The Activity Domain Theory conceptualizes knowledge integration as the task of achieving an optimal commonality penetration across cooperating activity domains. The case study from ABB indicates that two failed projects exhibited too strong commonality ambitions. The third, successful project, utilized a strategy of withdrawing some
commonality penetration while simultaneously pursuing forcefully other aspects of commonality.

Our findings indicate that enforcing an intrusive commonality conflicts with the internal ideologies of the activity domains. Debates and disagreements over the ‘proper way of doing’ things tend to slow down and even inhibit knowledge integration. The autonomy of the domains has to be adequately respected. Moreover, means of enforcing the necessary commonality, such as the dependency diagram, need to be easily understood, and aligned with the activity modalities and their interdependencies. One-dimensional diagrams showing only the temporal dimension, such as Gantt schemes are not sufficient. Based on these results, we suggest that the Activity Domain Theory may open up previously untrodden paths for theorizing about and intervening in knowledge integration enterprises.

References
Blackler F (1995) Knowledge, Knowledge Work and Organization: An Overview and Interpretation, Organization Studies, 16(6), 1021-1046
Grant, R. M. (1996b), Prospering in Dynamically-competitive Environments: Organizational Capability as Knowledge Integration, Organization Science, Vol. 7, No. 4, pp. 375-387


