A Self-Fueling Coordination Model for Enterprise Architecting Effectiveness

J. Alberto Espinosa  
Kogod School of Business  
American University  
alberto@american.edu

Frank Armour  
Kogod School of Business/  
CITGE  
American University  
frank.armour@att.net

Wai Fong Boh  
Nanyang Business School  
Nanyang Technological University  
awfboh@ntu.edu.sg

Mark A. Clark  
Kogod School of Business  
American University  
mark.clark@american.edu

Abstract

Enterprise “architecting” is the process of developing, updating and maintaining the EA. The goal of EA is to align business process and IT for the effective execution of business strategy. But the architecting process involves many stakeholders with very diverse and often conflicting goals. This makes architecting a daunting process. However, coordination in EA has received very little attention in the literature. A prior study reported that there are three fundamental types of coordination, each affecting architecting differently: (1) communication – the most prevalent, but least effective coordination mechanism in EA; (2) task programming – less prevalent but a very necessary type of coordination for EA success; and (3) team cognition – less understood but the most critical predictor of EA coordination success. In this paper we report on two case studies that investigated one highly successful and one less successful EA practice. Both organizations had competent business, IT and EA staff and active architecting coordination via communication and task programming. Only the successful organization, however, had strong team cognition, underscoring the importance of cognitive coordination in architecting.

1. Introduction

Enterprise architecture (EA) describes and documents relationships among business and management processes and information technology [20]. The main goal is to align business process with its underlying data, applications and technology infrastructure [4] to build applications in a more coordinated fashion, rather than as stovepipes, to support the execution of business strategies [36]. “Enterprise “architecting” is the work that people do to develop, manage and update the EA [12].

EA has many potential benefits like fewer system redundancies, faster system development, increased scalability and agility to cope with change. However, the benefits of EA can only be realized if the EA effort is well coordinated. For example, EA standards and governance practices can have detrimental effects if system developers find them too constraining or if the business process managers fail to see the connection between the EA and the processes they oversee. Research has shown that the level of EA maturity varies widely from organization to organization [36]. EA maturity can range from no EA (i.e., applications are stove piped) to one that relies on repeatable and modular elements that can be easily configured as conditions change. Moreover, many organizations do not stay static at one maturity level, but strive to move to higher maturity levels, which require substantial amounts of planning and coordination to maintain and update existing systems as the EA changes. Hence, most architecting takes place in the context of existing and often complex systems that are difficult to change. Consequently, the architecting process is replete with coordination challenges.

We argue that architecting is an inherently complex and multi-functional activity with multiple interdependencies. These interdependencies exist because of the need to align business and technology in a context in which old needs have to co-exist with the new. Managing interdependencies is, by definition, what coordination is all about [26-27], which leads us to posit that coordination in enterprise architecting is critical to EA success. While some have started to look into these issues [4, 13], there has been virtually no empirical investigation into how successful EA teams coordinate their work. Consequently, our research investigates the following research question:

What are the most effective coordination approaches in enterprise architecting?

In this paper, we first briefly discuss the enterprise architecting task and then provide the theoretical foundations of the study. Subsequently, we describe the study method and sample. We then discuss each of the two cases we analyzed. Finally, we discuss our findings and provide remarks.

2. The Enterprise Architecting Task

The main goals in EA are: (1) to create a tight alignment among business processes, data, software applications and the technology infrastructure in support of the business’ goals and strategy; and (2) to develop a high level of standardization and modularity in the EA components to implement individual systems more efficiently and become more agile to cope with change [36]. While the goals of EA are noble, the experiences with actual implementations have been mixed. A recent survey revealed that many EA implementations have been incomplete or unsuccessful
That same survey identified business-related and people-related issues as the main reasons for EA failure. In order to achieve tight alignment among the various EA layers, people working within these layers need to coordinate their work activities and align their goals. This, however, can be a daunting task because of the multi-functional nature of EA, often leading to conflicting goals among participants.

There are various EA frameworks describing these layers — e.g., Zachman, Federal EA Framework (FEAF), The Open Group Architectural Framework (TOGAF) [3]; most frameworks include four main layers: (1) business process; (2) data; (3) application; and (4) technical infrastructure. Because of the tight dependencies among these layers, the activities needed to define, implement, manage and change these four layers require a substantial amount of coordination.

Furthermore, an organization’s EA is not static; rather it changes over time as technologies change or as organizations move to more sophisticated levels of EA (e.g., from stovepipes, to standardization, to modularity) [36]. This dynamic nature of EA imposes severe coordination challenges because old and new processes and technologies have to co-exist and evolve in a synchronized fashion.

It should be obvious from this discussion that coordination is critical to effective architecting. When an EA program is put in place, the task activities of architects, IT staff and business stakeholders become interdependent. Managing these dependencies is no easy task, which is where coordination comes in [26-27]. Therefore, we draw on coordination theory to lay the foundations for our study.

3. Theoretical Foundations

Coordination has been defined as the management of dependencies among task activities [26]. Technical tasks are inherently interdependent [19], but most research associated with coordination in technical tasks has been conducted in the context of small teams [15, 18]. Very little research has been conducted in the context of larger scale technical collaborations among multiple individuals with diverse functional goals and perspectives, in largely asynchronous tasks in which time-to-completion is less of a concern, but the quality of the work outcomes is critical to success [13]. Examples of these types of tasks include things like strategic plans, technical platform overhaul, and enterprise architecting.

The traditional organizational literature suggests that teams coordinate organically and mechanistically [28, 38]. Organic coordination, also referred to as coordination by feedback or mutual adjustment, refers to coordination through communication and team interaction. Both formal (i.e., planned) and informal (i.e., spontaneous) communication have been found to help coordination in technical teams [23, 33], particularly with less routine aspects of the task. We posit that coordination in architecting requires a substantial amount of organic coordination because of the amount of information that needs to be exchanged, not only between EA layers, but among all participants to cope with change and to resolve conflicts that arise as a result of change.

Mechanistic coordination, also known as coordination by program or by plan, refers to the use of things like artifacts, processes, routines, project plans and schedules, specifications, workflow and procedures, to coordinate the most predictable and routine aspects of the task with minimal communication. Recent research has found that mechanistic coordination is beneficial for technical tasks, especially when communication is hampered [14]. We posit that coordination in architecting requires a substantial amount of mechanistic coordination in the form of standards, governance, implementation plans, EA change road maps, etc.

While the organizational literature on coordination is very useful, it is silent about cognitive coordination. The team cognition literature suggests that teams coordinate implicitly via mechanisms like shared mental models and team awareness [6], but this literature is generally silent about the role of organic and mechanistic coordination, except for general arguments made about the positive effect of team cognition on team processes (e.g., communication) [29]. Team cognition research suggests that team members coordinate implicitly when they have knowledge or schemas in common that can help them anticipate things [7, 21]. This is also referred to as “implicit coordination” [42] because team members coordinate based on silent assumptions about what others are supposed to do. Research has found that team cognition helps teams coordinate, but most studies have been conducted with real-time and often simulated tasks [6]. There is very little research on the effects of team cognition on coordination in larger teams working on asynchronous, long-term, complex or multi-functional tasks [15]. We posit that the benefits of team cognition on coordination generalize to such asynchronous tasks like architecting; indeed, cognitive coordination may be more important for such tasks as contextual cues typically present in synchronous communication are absent.

There are many different team cognition conceptualizations and constructs defined and studied in the literature [6]. We focus in this study on a few constructs associated with long-term knowledge, which are more applicable to coordination in long-term, asynchronous tasks like EA, rather than those associated with fleeting knowledge (e.g., situation awareness, team awareness), which are more
applicable to fast paced, real time tasks [10]. We briefly define these team cognition constructs next.

Shared task knowledge is knowledge that collaborators have about the tasks that others are carrying out [21], which can help them plan their own activities more effectively. Because the tight alignment of business processes, data and technologies is a key goal of EA, we posit that it is paramount that business people, EA staff and IT staff understand each others’ task domains sufficiently for their respective activities to be effectively coordinated.

Shared mental models refers to the alignment, similarity or overlap of individual members knowledge and mental schemas [34-35]. Shared mental models go beyond shared knowledge because it also incorporates similarities in the mental schemas that members have about the task and about each other, which helps them explain the actions of others and anticipate how they are likely to behave, thus making the team more coordinated [21]. Shared goals, shared beliefs and shared vision are examples of shared schemas that help groups push in the same direction towards a common goal. Shared mental models are particularly important in architecting because the main goal of EA is to align business processes, data and technology to provide consistent support to execute business strategies.

Transactive memory is knowledge members have about what other members know, which is useful to: locate expertise when needed; know where to go for specific information; and know where to channel new information [39-40]. Expertise coordination [16] is a similar concept, which also includes knowing when to bring a particular member’s expertise to bear into the task. Because of the inherent dependencies associated with the functional diversity in EA, we posit that transactive memory is important in architecting.

Common Ground is the mutual knowledge or knowledge members share and know they share [11, 22], which is manifested in the form of shared vocabulary and mutual understanding of the terminology used [9]. Lack of common ground leads to miscommunication and confusion because members associate different meanings to the same terminology. We posit that common ground is particularly important in architecting because there are two languages that need to be understood: business and technical, in addition to the multiple functional areas or domains that may be involved with architecting.

Collective mind, also referred to as collective sense making, is perhaps the most powerful form of team cognition when it comes to coordination. For example, it has been found to be key to operating safely and reliably in extremely risky environments (e.g., nuclear aircraft carriers) [41]. Collective mind requires heedful interrelating in which individuals understand and purposely take into account how their actions will affect others in the team, which is critical in high risk environments. But it can also be beneficial in less risky situations like every day vehicle driving in that one not only needs to drive well and anticipate what others are likely to do, but also how our driving will affect others on the road. We posit that collective mind is very important in architecting because architects need to fully understand and take into account how their decisions and actions will affect the work of business stakeholders and IT staff, and vice versa.

In sum, research on team coordination has only recently begun to integrate organic, mechanistic and cognitive coordination perspectives; their interplay is still unclear. For example, a recent study found that team cognition affects architecting effectiveness directly, but it does not influence organic or mechanistic communication directly. Rather, it moderates the effect of organic and mechanistic coordination on architecting effectiveness [13]. Furthermore, the study found that these are only the static effects; team cognition gets stronger over time as architecting effectiveness and organic and mechanistic coordination become stronger – i.e., a “self-fueling” dynamic temporal effect. We have adopted selected parts of the model used in that study, which are applicable to the present study. This revised research model is illustrated in Figure 1.

![Figure 1: Self-Fueling EA Coordination Model](image)

The solid lines in this model represent the static effects at a point in time. They show that organic, mechanistic and cognitive coordination directly influence architecting effectiveness; and cognitive coordination also moderates the effect of organic and mechanistic coordination on architecting effectiveness. The dotted lines represent the dynamic effects over time, which create the self-fueling effect – i.e., as the enterprise architecting becomes more effective, all forms of coordination become stronger; and stronger organic and mechanistic coordination, in turn, enhance cognitive coordination. Consistent with Grounded Theory, we do not elaborate further on this model, rather we use it as a reference to guide our analysis. Hence, we are not using Grounded Theory for theory development, but only as a data analysis method.
4. Study Method and Sample

Our empirical study is based on qualitative semi-structured interviews, which have been widely used for exploratory studies in information systems research [24-25, 30-31]. We use a case study approach, which is appropriate in exploratory and descriptive studies in which the work context matters and multiple cases are appropriate to contrast results [43]. We contrast two case studies from two organizations with very different experiences with their EA.

Because EA involves the alignment of business and IT, there are many individuals with diverse functional backgrounds who either work in EA functions or are affected by the EA. Therefore, and consistent with prior research that has identified key architecture roles [4], we interviewed five participants in each organization: the Chief Information Officer (CIO); the chief enterprise architect; a technical architect; an IT staff affected by the EA, but not a member of the EA group; and a business stakeholder affected by the EA. These two organizations were selected because they are both seriously committed to their EA programs and have very competent architects, IT staff and business staff with many years of relevant experience. However, one organization has been widely successful in their EA implementation, whereas the other one is struggling. We posit that important differences between these two organizations include their approach to architecting and how their EA activities are coordinated, and that these influenced the respective outcomes.

The interview instrument we used is semi-structured with questions to elicit information about coordination challenges, interdependencies among functions and best practices. The interviews were audio recorded and transcribed verbatim, yielding approximately 7 hours, 120 pages and 56,000 words of interview material. The interview transcriptions were first analyzed one at a time using the Grounded Theory approach [17, 37], then analyzed by recurring themes [8], and once more by case (i.e., by organization) [43]. Grounded Theory is a widely used qualitative method in information systems research [5, 31] and global collaboration [32]. As prescribed by Grounded Theory, each case was analyzed to interpret similarities and differences among the two cases.

Consistent with Grounded Theory, we developed an initial (manual) coding scheme through open coding of the first few interview transcriptions to uncover general recurring themes of interest, with consensus from 3 of the researchers involved in this study. We then refined this coding scheme using NVivo© software, using axial coding of the text data to find relationships among these themes. Again three researchers coded the data independently with similar results, then interpreted the results jointly. Finally, we analyzed each case individually to contrast their differences. We now discuss each of the two cases.

5. Case #1 - OWM

Background. This organization is a private company specializing in oil well management (OWM) with about 1200 employees. It was established in 1997 by a large oil company. When OWM got started, management decided to adopt an EA approach. They spent four to five years on planning before implementing their EA strategy. The strategy was implemented first around business process modeling, then around data modeling and finally around application and technology infrastructure architecting. After 5 years, they separated the business process aspect of EA from under the CIO organization and moved it to a new unit called the Center for Process Excellence (CPE). The implementation planning took 6 months and it involved 32 business process owners.

Organization. The CIO and the Vice President for the CPE both report directly to the CEO and both departments have a very tight cooperative relationship. The CPE keeps the corporate standard procedures and work processes for the entire company and is responsible for business process architecting. The EA functions for the remaining three layers (i.e., data, applications and technology) are managed by the chief enterprise architect, who reports to the CIO. The CIO also supervises all other IT functions. The intent is for CPE to identify and inventory the best processes and for the CIO to deliver the supporting technology to support those processes. The current EA maturity level of this organization is quite high, somewhere in the early stages of level 4 (modularity) of the Ross et al. maturity model. The 4 EA layers appear to be tightly aligned and all interview participants were consistent in their high EA maturity rating of the organization.

Architecting. OWM identified early on that information and business processes cannot be separated and it became clear that the CIO’s group and CPE have to be tightly coordinated. All IT positions are integrated into data, applications or technology architecture groups, all reporting to the chief architect, who in turn reports to the CIO. It was recognized early on that an organization had to be created to provide a view of the entire enterprise, from a process perspective. The philosophy in this group is that EA is a way of doing business and employees have adapted to this new way of doing things. It was a costly long term program to get the EA in place, but the implementers had support from the CEO who understood the value of the investment in EA. Top leadership and commitment are key to the EA effort. As one participant commented, “‘I’m talking about a top leader that's holding the flag, running, leading the group up the hill.’
**Organic Coordination.** OWM spent a substantial amount of time on one-on-one communications. The main purpose of the communication activities is not so much to coordinate day to day activities, but to share information, knowledge and view points to develop shared cognition (e.g., shared goals, common ground). As one participant stated: “we have the process folks in the room with the line management and leadership, and they are all on the same page...our job is to facilitate the conversation with managers and have them work with us to design the best way to do oil treating at any oil field.” A lot of the IT/business alignment comes from proactive one-on-one discussions with subject matter experts, who are typically from the business side, rather than from the IT side. All conflicts were quickly addressed and discussed to find the best resolution.

**Mechanistic Coordination.** OWM spent five years planning their EA implementation and placed a lot of emphasis on standardization. They spent a substantial amount of time finding similarities in processes in their various units to identify best practices and develop standardized, replicable processes. This required some of the existing processes to change, which was a hard sell at the beginning. They also developed meta data models and data directories to help anyone get to the right data very quickly and have a common frame of reference. Data is used as a source of common ground, as one participant stated: “one of the hallmarks of a good enterprise architecture is to have a very well understood enterprise data model.” OWM also relied on pro-active governance as a coordination mechanism.

They established an architectural board responsible for reviewing the enterprise architecture projects and various milestones throughout the project, to ensure compliance with the EA. This board included the chief architect, the infrastructure, data person, and business staff, which reviewed project deliverables, and reported back to project teams to take corrective action when needed. Another aspect to governance as a coordination mechanism came from budgeting in the sense that any substantial project would only be prioritized and obtained IT resources if it was compliant with the EA.”

**Cognitive Coordination.** The most distinguishing feature of OWM’s EA was their high level of cognitive coordination, including sharing critical knowledge of the associated tasks, persons, and coordinating processes. OWM’s EA invested in procedures that ensured that there would be collective understanding of the company vision and operation. Getting the EA right required a lot of change over a long period of time and this was a very difficult thing to do. However, the top leadership in OWM understood the value of EA and supported the EA program, not just financially but also through leadership and commitment. One important philosophy of OWM’s EA program was the strong perspective that the main customers were the business process owners, not the IT developers. No process or system was allowed to be modified without the approval of the business process owners affected. This was labeled the “optimized ownership model.” Process owners had process analysts working with them, playing a bridging communication role between IT and business, thus helping to share knowledge and to establish common ground.

System projects were generally led by business managers, not IT managers, thus establishing a common frame of reference. Also, the EA effort involved a substantial amount of training with all the key people to help achieve commitment and a shared vision. While all systems development was driven by business process owners, the EA group spent a substantial amount of time training business stakeholders on EA concepts and practices.

CPE staff spent a lot of time travelling to different subsidiaries to facilitate exercises that ensured everyone understood the necessary terms, definitions and principles associated with the EA, thus establishing strong common ground and commitment (i.e., a shared schema) to the EA. Part of this discussion centered on helping employees understand that an enterprise approach was best for the company over the long run, even if it was not the most expedient approach for some groups. This effort to educate employees about the importance of EA helped to develop a shared mental model about EA, which paid off for OWM, as one participant expressed: “even the folks who work in IT didn’t have a good feeling about anything to do with architecture, and those same folks ... got such an appreciation for it now, they are the evangelist for it. It was just a complete turnaround and people’s perception of what the right thing is to do, and what’s better in the long term and short term.”

Cognitive coordination influenced OWM, not only at higher levels, but also at various technical layers. One such area in which common ground was formed was around data, as expressed by one participant: “a lot of it about bringing people together is really coming to agreement on a data point of view, what are these things and what are we going to call them.” A lot of time and effort went into creating standard data repositories with data element names that had common meaning for everyone who used the data and ensuring that the data collected met the needs of all associated data consumers. Because everyone in the company either produces or uses data, well defined and understood data models were instrumental in achieving common ground on EA.

Perhaps one of the most salient aspects of OWM’s architecting was the directed effort to develop a collective mind about EA. From the business
perspective, everything in the EA was driven by business process needs, so no IT decisions or actions were undertaken without understanding how these supported business process needs. And from an IT perspective, business stakeholders were trained on EA practices so that their actions and decisions were optimized with respect to the EA. This kind of heedful interleating ensured effective alignment between business and IT.

Summary and Other Findings. OWM’s architecting practices leading to EA success provides support for the “self-fueling” effect of cognitive coordination. We observed that all forms of coordination at OWM contributed positively to the effectiveness of their architecting efforts. However, before the benefits of EA were evident there was some early skepticism. Because the top leadership committed to the EA program, it was accepted by employees. As key managers worked to develop shared mental models, common ground, and collective mind about EA, the benefits of EA (e.g., standardized data models, repeatable business processes, less redundancy and overlap, common technical language) started to materialize. In this way, an effective EA becomes a mechanistic coordination artifact per se. As one participant stated: “a comprehensive EA provides quality information to everyone in the organization who needs it and they all become better. We have better HR, we have better finance and better engineering and that’s what makes a great company.”

Another participant also stated that “the main impact EA has is that it forces you to operate in terms of a whole; you cannot operate within a silo.” This increased architecting effectiveness not only made organic and mechanistic coordination more effective, but also strengthened the group’s collective cognition in a dynamic self-fueling cycle. As one participant stated: “looking back, it’s easy to see the benefits. It’s easy to see what a data warehouse looks like and all the fantastic things users can do with the data that’s in there and how integrated it is.”

6. Case #2 – IRO

Background. This organization is an independent regulatory organization (IRO) overseeing and providing continuing education services to over 4500 firms, a number of institutions and over 600,000 individuals within those organizations. The organization has over 3000 employees, and has been in business over 70 years. The organization’s products are primarily information and tools, thus application development is a key business activity.

Organization. The organization has 6 lines of business (LOB), a Product Management group and the CIO’s group. The chief architect reports to the CIO and oversees and steers the EA, but does not have ownership over the architects or applications. He is responsible for the oversight and direction of shared services and shared frameworks. The EA function is primarily concerned with data, software and technology architecture for business applications, but does not have any role in business process architecting. Business requirements analysis is done by a group of Product Management staff who work closely with technical project managers and application developers, but not with EA staff.

The application architects report to an application manager, who oversees all applications. Some architects have dual responsibilities, one as development managers for a few projects and another for EA oversight of other projects. This semi-matrix management approach was adopted to have multiple eyes on various aspects of system development to achieve high product quality. What is called the “software factory” includes functions like requirements management, quality assurance, configuration management, and infrastructure. The servers where applications run are managed by an external service provider, so the operational groups are more involved in managing this outsourcing relationship than in managing the infrastructure.

Architecting. There is some segmentation in the EA with “tower” architects who support applications for specific LOBs. The chief architect coordinates, but does not supervise the work of the tower architects. The tower architects work with application developers to provide support for data, software and technology architecture issues. Product managers carry out the business requirements analysis function, bridging business functions and EA and IT (i.e., business does not work directly with EA and IT). In addition there are technical requirements analysts who deal with the technical systems. The architecting for IRO is more focused on technical architecture aspects like: service oriented architecture; data warehousing; etc. The EA maturity of IRO is fairly low, relying primarily on standards and project teams to keep things organized, but the LOBs operate somewhat independently as stovepipes to some extent, and most of the architecting is done for individual applications.

Organic Coordination. There is a fair amount of coordination via one-on-one communication, but most of it is ad-hoc as needed. The most intensive communication happens between product or application managers and application architects. There is very little communication on EA issues with the chief architect or across LOB’s, except for enterprise-wide EA components. But coordinating organically beyond the immediate collaborators, which is essential to EA can be a challenge, as exemplified by this comment from a participant: “it can take... getting a meeting set up that requires multiple senior managers to be in it, given that their calendars are almost 100%
booked all the time. It’s very difficult to get those lined up. The pace of meetings – it might take months to get something that really needs only 3 or 4 meetings to get accomplished.”

**Mechanistic Coordination.** There are some EA standards and central data repositories in place, but these are not necessarily adhered to. EA governance is not pro-active, but reactive in the form of audits and sometimes are even ad-hoc in nature. While EA reviews are meant to ensure compliance with the EA, waivers are often granted because the resources for development are controlled by the business units. For example, for data that spans more than one LOB, each LOB generally keeps a separate copy of the data for their own needs, rather than sharing a common data repository. In addition, some groups that could be sharing data do not do so. A data hub was recently established to address this issue, but it was implemented more as a central data warehouse, rather than live shared date, but at least this warehouse can now be used by different applications for validations, lookups, and displays. However, the group that owns this data is not part of the application development group, which creates coordination challenges.

Even for the technical layers, some standards are not tightly enforced. The result is a mix of technologies like multiple database and software development platforms. Also, when changes in the application architecture are required, the developers coordinate this organically with EA, instead of coordinating via a formal architectural review process. Business stakeholders participate in steering committees for projects and applications, but not in architecture steering committees, as one business stakeholder commented: “most of the EA work happens outside of the areas that I’m responsible for. So you know, I… to be honest with you, I really don’t deal with it.”

**Cognitive Coordination.** There was not much evidence of shared vision, shared mental models or commitment around EA among stakeholders, based on responses by IRO interviewees. The system needs of the LOBs generally drive the development effort and many similar processes are supported by different systems. EA is viewed by product managers, application managers and technical architects as a technical endeavor supporting specific applications.

Business stakeholders have very little shared knowledge with enterprise architects and they rely on their technical project managers and technical architects to work with the EA group. There is a shared belief among business stakeholders and application developers that EA is a technical activity, as this comment made by a business stakeholder illustrates: “my guess is the technical project manager does interface with enterprise architecture much more than I (do) because they are both in the software development factory, so there’s probably a lot that goes on there that I am not aware of.”

Similarly, there is very little in the form of shared schemas about repeatable processes or overlapping EA components across LOBs, which function as stovepipes. The EA function is making an attempt to take a centralized approach to systems development to leverage on process and data similarities, but the focus is on data and infrastructure, with very little enterprise-wide architecting at the business process and software levels. IT/Business alignment is tight, but it is accomplished at the applications development level, rather than at the EA level. When the business processes or IT do not align with the EA, the EA yields. For example, if a particular off-the-shelf application meets the needs of a particular LOB it will be acquired regardless of EA standards.

Finally, common ground was lacking between EA staff and other staff. For example, the EA discussed the concept of “tower architects” who are the architects supporting particular LOBs, but this term did not have a meaning to other participants in the study.

**Summary and Other Findings.** One coordination challenge has to do with the budgeting and resource allocation, as evidenced by this comment from one participant: “we start to identify opportunities to participate in EA, the reaction often times is, that’s great….we understand, but we’re locked in, and we’re already committed to a budget and schedule to deliver these things for this year…there is very little flexibility to absorb anything else that could impact us, no matter how much it is for the common good…the challenge then for EA perspective, is to try and herd this people together during that planning cycle.”

In sum, this organization has very competent architects, application developers and business stakeholders who are very committed to product quality. There is also a good vision for the EA articulated by the chief architect. But this vision is not shared by the business stakeholders and application developers, who view EA more as a technical support issue dealing primarily with infrastructure and data. EA is viewed more as a compliance issue, which is generally waived when business stakeholders make their case. There is tight organic coordination, but primarily between business stakeholders and application developers, otherwise communication is ad-hoc. There is some mechanistic coordination, but the adherence to standards and processes associated with EA is limited. Finally, there is very little cognitive coordination in that the business stakeholders and application developers do not quite understand the role of EA and, conversely, the EA has very little involvement with business processes. This weak cognitive coordination has thus failed to ignite the self-fueling mechanism that can make EA more effective.
7. Discussion of Results

Our results, summarized in Table 1, support the self-fueling model of coordination for enterprise architecting effectiveness. Our results provide preliminary empirical evidence that cognitive coordination is the “fuel” that strengthens over time dynamically. Enterprise architecting effectiveness requires organic, mechanistic and cognitive coordination. But strong cognitive coordination strengthens the effectiveness of organic and mechanistic coordination. But perhaps the most interesting part of our results is that the EA itself is a powerful enterprise coordination mechanism and as stakeholders begin to experience the tangible benefits of EA they become believers. As shared vision, shared mental models, common ground, collective mind and other forms of cognitive coordination strengthen, all other coordination mechanisms become more effective, thus improving architecting effectiveness over time.

In sum, good organic coordination (i.e., effective communication) and sound mechanistic coordination (i.e., effective processes, tools and routines) are both important for effective coordination in enterprise architecting. However, as we saw in the IRO case, these are necessary, but insufficient ingredients for

<table>
<thead>
<tr>
<th>OWM</th>
<th>CPE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Architecting Approach</strong></td>
<td></td>
</tr>
<tr>
<td>The CPE deals with business process and</td>
<td>Somewhat similar to OWM in that the LOBs</td>
</tr>
<tr>
<td>nothing is done without the approval of</td>
<td>own the processes and work tightly with</td>
</tr>
<tr>
<td>business process owners who work in</td>
<td>the architects, but the architects report</td>
</tr>
<tr>
<td>tight collaboration with architects; EA</td>
<td>to the application manager and work with</td>
</tr>
<tr>
<td>reports directly to the CIO and handle</td>
<td>application developers, with little</td>
</tr>
<tr>
<td>application, data and infrastructure</td>
<td>interaction with the EA group; the EA</td>
</tr>
<tr>
<td>architecture; strong emphasis on</td>
<td>group serves as a coordinator, but does</td>
</tr>
<tr>
<td>repeatable EA components and the EA</td>
<td>not get involved with business processes;</td>
</tr>
<tr>
<td>rather than individual applications. EA</td>
<td>strong emphasis on individual applications</td>
</tr>
<tr>
<td>is driven by business.</td>
<td>rather than the EA and little emphasis on</td>
</tr>
<tr>
<td></td>
<td>repeatable EA components. EA is viewed</td>
</tr>
<tr>
<td></td>
<td>as a technical endeavor.</td>
</tr>
<tr>
<td><strong>Organic Coordination</strong></td>
<td></td>
</tr>
<tr>
<td>Substantial amount of communication,</td>
<td>Tight communication within specific LOBs</td>
</tr>
<tr>
<td>especially one-on-one; most of it aimed</td>
<td>and for specific applications;</td>
</tr>
<tr>
<td>at developing shared cognition about EA.</td>
<td>communication between business or</td>
</tr>
<tr>
<td></td>
<td>application developers with EA staff is</td>
</tr>
<tr>
<td></td>
<td>minimal and ad-hoc.</td>
</tr>
<tr>
<td><strong>Mechanistic Coordination</strong></td>
<td></td>
</tr>
<tr>
<td>Spent several years planning the EA</td>
<td>Some standards and procedures are in</td>
</tr>
<tr>
<td>implementation; intensive reliance on</td>
<td>place, but not always followed, and it is</td>
</tr>
<tr>
<td>standards, processes and centralize data</td>
<td>not that difficult to get waivers.</td>
</tr>
<tr>
<td>models.</td>
<td></td>
</tr>
<tr>
<td><strong>Cognitive Coordination</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Shared Task Knowledge</strong></td>
<td></td>
</tr>
<tr>
<td>Substantial amount of knowledge sharing</td>
<td>Substantial amount of knowledge sharing</td>
</tr>
<tr>
<td>between business process owners,</td>
<td>within applications, but not across</td>
</tr>
<tr>
<td>developers and architects.</td>
<td>LOBs or with EA.</td>
</tr>
<tr>
<td><strong>Shared Mental Models and Shared Schema</strong></td>
<td></td>
</tr>
<tr>
<td>Very strong shared schemas about the EA</td>
<td>Weak shared schemas about EA; shared</td>
</tr>
<tr>
<td>and the benefits of EA, developed through</td>
<td>schemas are mostly within LOBs and</td>
</tr>
<tr>
<td>extensive interaction and a substantial</td>
<td>specific applications, but not across</td>
</tr>
<tr>
<td>amount of upfront planning.</td>
<td>LOB's or with the EA.</td>
</tr>
<tr>
<td><strong>Transactive Memory</strong></td>
<td></td>
</tr>
<tr>
<td>Strong knowledge of who knows what, both</td>
<td>Strong knowledge of who knows what</td>
</tr>
<tr>
<td>by names and by roles, and a solid</td>
<td>within LOBs and applications, but less</td>
</tr>
<tr>
<td>familiarity with the work responsibilities</td>
<td>so across LOBs and with EA staff.</td>
</tr>
<tr>
<td>associated with roles.</td>
<td></td>
</tr>
<tr>
<td><strong>Common Ground</strong></td>
<td></td>
</tr>
<tr>
<td>Strong common ground purposely developed</td>
<td>Little evidence of common ground across</td>
</tr>
<tr>
<td>through training and planning sessions to</td>
<td>LOBs, and between business or developers</td>
</tr>
<tr>
<td>develop common understanding on vocabulary and terms.</td>
<td>with EA staff.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Collective Mind</strong></td>
<td></td>
</tr>
<tr>
<td>Very strong collective mind achieved</td>
<td>No collective mind observed, except</td>
</tr>
<tr>
<td>through training by the CPE about how</td>
<td>within LOBs and applications.</td>
</tr>
<tr>
<td>business process, data, applications and</td>
<td></td>
</tr>
<tr>
<td>technology infrastructure affect each</td>
<td></td>
</tr>
<tr>
<td>other and about how decisions within</td>
<td></td>
</tr>
<tr>
<td>these impact the effectiveness of the</td>
<td></td>
</tr>
<tr>
<td>overall EA and vice versa.</td>
<td></td>
</tr>
<tr>
<td><strong>Self-Fueling Effect</strong></td>
<td></td>
</tr>
<tr>
<td>The benefits of EA are observable and</td>
<td>No evidence of self-fueling effect</td>
</tr>
<tr>
<td>evident, which has helped strengthen the</td>
<td>observed due to the weak shared cognition.</td>
</tr>
<tr>
<td>shared cognition of those involved with</td>
<td></td>
</tr>
<tr>
<td>architecting, which has made organic and</td>
<td></td>
</tr>
<tr>
<td>mechanistic coordination more effective.</td>
<td></td>
</tr>
<tr>
<td>The self-fueling effect and its benefits</td>
<td></td>
</tr>
<tr>
<td>are apparent.</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Summary of Results
EA success. In contrast, as we saw in the OWM case, cognitive coordination provides three additional types of effect that are critical to EA success by: (1) directly affecting architecting effectiveness (at any given point in time); (2) positively moderating the effect of organic and mechanistic coordination on architecting effectiveness (at any given point in time); and (3) progressively strengthening all forms of coordination and EA effectiveness over time (i.e., a self-fueling effect).

8. Conclusions

Our study provides contributions for both research and practice in two areas, EA and group cognition. Our contributions are important because EA is a long-term activity that requires the collaboration of people from multiple functional backgrounds with very diverse goals and objectives. Moreover, by definition, EA seeks alignment between business and technology, thus making the development of shared cognition critical in successful architecting. This represents an important contribution to the EA literature because most prior studies have focused on technical [2, 20] or governance [4] aspects of EA and our study clearly show that effective coordination of the architecting effort is critical to EA effectiveness. Our study also contributes to the shared cognition literature because most prior studies in this line of research have focused on small teams working on short-term, real-time tasks. Our study shows the applicability and benefits of shared cognition in larger collaborative effort in multi-functional, long-term, asynchronous tasks like enterprise architecting.

The interesting thing with respect to shared cognition research is that there seems to be some degree of consensus in that literature that shared cognition has a direct effect on team process (e.g., team communication), which in turn has an effect on team performance [6, 29]. This is a natural conclusion for real-time teams in which effective communication is key to coordination and performance, and that the effect of team cognition on performance is mediated by team communication. We find evidence in our study that these prior findings do not generalize to long-term teams working in asynchronous tasks because communication has less direct bearing on the success of the task, but on the other hand, communication is critical to building strong team cognition, which in turn can make communication more effective. This type of dynamic, self-fueling effects (i.e., communication being both, an antecedent and a consequence of shared cognition), have not been previously investigated.

Our findings are important for practitioners because they provide specific guidance about the need to build shared cognition as a cognitive coordination mechanism to help the organization achieve a self-fueling state for effective architecting. With respect to shared cognition, our study underscores the importance of being able to assess the level of shared cognition within this group. This research has provided the motivation for a subsequent study underway by some of the authors to provide the conceptual and theoretical foundations to represent and measure shared cognition in groups. This will be done by developing fundamental dimensions of team knowledge and social network analysis methods to measure, analyze and visually represent shared cognition in large groups.

9. Acknowledgements

This study was financed by the Center for Information Technology and the Global Economy (CITGE) at the Kogod School of Business at American University, Washington, D.C.

10. References
