Coordinating Distributed Operations

Daniel Oppenheim, Saeed Bagheri, Krishna Ratakonda, and Yi-Min Chee

IBM Thomas J. Watson Research Center, Hawthorne, NY 10562, USA
{music,sbagher,ratakond,ymchee}@us.ibm.com

Abstract. In this manuscript, we discuss the need for, and present a new system architecture for cross collaboration among multiple service enterprises. We demonstrate the importance and inevitability of such collaboration along with challenges in its proper realization through several real-life examples taken from different business domains. We then show that these challenge are rooted in two key factors: unpredictability and responsiveness. The key contribution of this manuscript is the presentation of a new model, centered on an intelligent hub, for coordinating the logistics of cross enterprise collaboration. This hub is constructed in a manner intended to directly identify and solve the two key fundamental challenges of cross enterprise collaboration. As such, we expect it to outperform other means of collaboration across service providers. We demonstrate the potential for such performance using field examples.

Keywords: service operation management, service quality, logistics, enterprise ecosystem, globalization.

1 Introduction

Service operation and management is becoming increasingly complex as the *doing* of work shifts from within the enterprise to a distributed ecosystem of service providers. While this new ecosystem provides a desirable flexibility [1,2], it challenges both the sustained effectiveness of operations and the quality of delivered services. It has accordingly become apparent that a new approach is required to address this challenge [3]. Using such an approach service operation management will become *aware* of the ecosystem and have the ability to optimally manage the logistics of its inherent complexity. Moreover, a business could then manage its operations while reasoning within its own business terms. Such optimal operation management will allow us to realize the potential benefits of the ecosystem while maintaining an acceptable level of quality.

Forces of globalization and the ever growing need for differentiation and innovation are moving work from a co-located to a distributed environment. Companies form collaborations to achieve a goal that none could achieve individually. They all share the financial burden and development risk, but benefit from the sum of their differentiating capabilities. We call this new model of doing business Cross Enterprise Collaboration (CeC). For example, Airbus is developing its new Airbus-A380 in a consortium with several partners, and similar trends

E.M. Maximilien et al. (Eds): ICSOC 2010 Workshops, LNCS 6568, pp. 213–224, 2011.

can be observed in all industries. However, collaboration, while inevitable, is coming at a huge price. The rate of project failures is alarming. The delivery of the Airbus-380 was two years overdue with accumulated losses exceeding 6B USD [4]. This breakdown, while extreme, is common across different industries, including software and automotive. Clearly, there is a need for a new approach to manage cross enterprise collaboration.

The problems of CeC are well recognized by industry. A study made between 2007 and 2008 by the Aberdeen Group [5] examined more than 160 enterprise products that required collaboration between mechanical engineering, electronic engineering, computer engineering, control engineering, and system design. 71% of the people interviewed identified a need to improve the communication and collaboration across the silo disciplines; 49% required improve visibility; 43% raised the need to implement or alter new product development processes for a multidisciplinary approach. The primary reason for breakdowns is the paucity of support for coordinating the overall work between the silo disciplines.

We approach the problem from two complementing directions. First, we model the ecosystem of the enterprise collaboration. This provides the context in which a business can identify the main actors and reason about how they relate to each other: organizations, people, process, and work. Second, we use service oriented concepts to encapsulate the *doing* of work from managing and *coordinating* how it gets done. This encapsulates each service provider in a space separate from the rest of the ecosystem. Accordingly, only the necessary and sufficient information of the ecosystem needs to be communicated in order to coordinate a provider. Once this is done, the complexity of Business Process Management (BPM) suddenly reduces to many independent sub-BPM problems that are both known and tractable. The challenge, however, is formation of the appropriate encapsulated space for each service provider and definition of the information flow. Another way to think of this encapsulation is as an extension of the concept of Work as a Service (Waas), where any domain work be provided as a service and carried out by any qualified provider organization. A key component of the model is dedicated specifically to support this: a hub. In a companion paper we provide detailed analysis of hub operation and show how an optimum policy can be determined. This policy, among other things, describes how service requests are distributed amongst providers [6]. The model we present here presents a broader view and addresses more directly business needs and concerns. The model might be viewed as a significant generalization of the Distributed Enterprise Services pattern [7,8], and incorporates Malones theory of coordination [9] and work [10].

In the next section we illustrate the core challenges facing a proper implementation of CeC. In particular, we discuss the role of unpredictability and how it affects the complexity of CeC. We further introduce a new perspective for decision making that is both holistic and domain aware. In section 3 we incorporate our new perspective in decision making into the structure of an intelligent hub and propose a new model to support CeC. We demonstrate the operation of our model through an illustrative example. The section after that reviews the key properties of our hub model such as agility. We conclude in section 5.

2 Collaborative Enterprise Work in Development

Product development is complex and is carried out in large projects. Projects require deep domain expertise in several disciplines, last from months to years, and are expensive; their success is critical to the business. Detailed processes, methodologies, and best practices guide the project lifecycle. However, plans can only form a common starting point because unpredictable events always happen. Issues may arise following a realization that a requirement is unclear or a design is faulty. Assumptions may prove to be wrong. Required materials may become unavailable or new technologies may make current progress obsolete. Partners or providers may come and go, and disasters certainly happen. Uncertainty and unpredictability can only be managed by flexibility and adaptation, which is why every project deviates from its original plan in unique ways.

Successful completion of development projects depends on containing their inherent unpredictability. To do this one must identify issues early, figure out the best response, and make the necessary changes. Containment in CeC work is too complex to automate. Were Airbus able to automate the creation of a blueprint for the A-380, it would. CeC work, therefore, has a critical dependency on human knowledge, experience, leadership, and creativity. People must do everything that technology cannot, and technology must support them in doing what is left. To understand what needs to be supported and how, we next examine the different roles of humans and organizations in responding to an unpredictable event.

2.1 The Three Perspectives: Executive, Operation, and Domain

Figure 1 depicts the three fundamental roles of people: executive, operation, and domain. The *executive* role is responsible for the enterprises strategy and business goals; it also pays close attention to the marketplace, competitors and customers. *Operation* is a managerial role that is concerned with efficiency, and meeting deadlines; it focuses on the coordination of work between the silo disciplines. *Domain* roles embody the competency and skills required to do the highly specialized work that is the underpinnings of the final product.

2.2 Containing Unpredictability through Exception Handling

Determining the best response to an unexpected event can be thought of more generally as *human exception handling.* The following scenario illustrates how the three roles relate to each other in the context of CeC work. Large development projects distribute work between several teams that do *requirements, design, build, test,* and *integrate.* In this scenario, the electric design team realizes that it may be two month overdue. It wants to determine how serious the problem is so that it can decide what to do about it.

The seriousness of the problem can range anywhere from having no impact at all on the project to killing the enterprise by giving a competitor a two month advantage. The team cannot possibly determine which because it lacks visibility into the operation and executive domains.

What is a reasonable response? The team may decide to reallocate resources from other tasks in order to overcome the anticipated lateness. However, this may be the worst thing to do. For example, it may turn out that the root cause stems from an engineering error in the engine's design. If such is the case, the engineering team should first fix its design before anyone takes further action, as the error may affect all ongoing and future work. Operations, on the other hand, have different concerns and will likely make a different decision. If, for example, meeting the project deadline is a key priority, then operations may insist on meeting the original deadline, well aware that quality will be seriously sacrificed. On the other hand, an executive perspective may lead to a very different approach: termination of the entire project at a loss and reallocation of its resources to a more profitable venture.

All the approaches described above are reasonable, but each reflects a different concern. To make an informed decision one must have the full context and balance all related concerns. Figure 1 depicts the creation of the required context by gathering information and concerns from each level. Information about the domain can be gathered from each discipline. Operations can provide relevant information relating to the project scope. The partnering executives can add relevant information from the perspective of goals, strategy, competitors, or the marketplace. But there may be additional information that is required: new technologies, advancements by a competitor, etc. This is why key stakeholders must take part in the decision making process: to voice their concerns and supply additional information and insight that can only come from human experience and expertise.

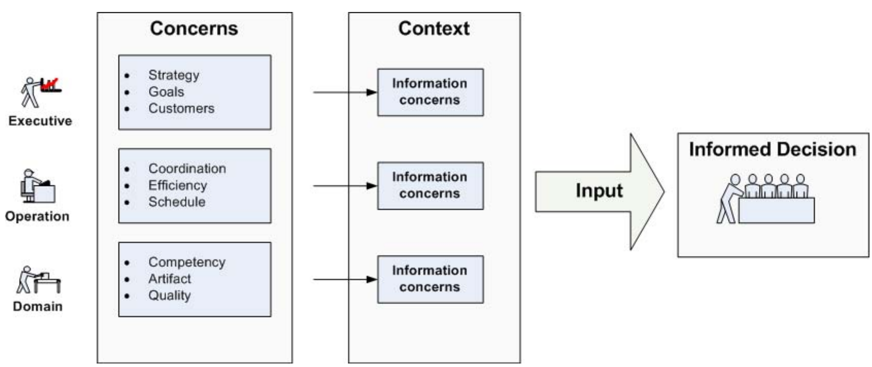

Fig. 1. When something unpredictable happens it must be contained. The sum of information that is gathered from all three domains provides a big picture. Based on this context the stakeholders can make an informed decision.

2.3 Patterns of Coordination

Figure 1 outlines a desirable *issue resolution* process. To enact that which was decided upon requires coordination. But coordination can be complex. If the root problem of the electric design team was a fault in an engineering design, then the need to fix the design can easily be communicated to engineering. However, other teams may also have dependencies on engineering and they must also be notified. So far, so good, but this is where complexity begins to compound. The changes that engineering make may require other disciplines to modify their designs so that they accommodate the changes which engineering made. Each new change in every other discipline may start a new cycle of updates and dependencies between all disciplines. This situation can quickly become unmanageable.

Figure 2 depicts two patterns of coordination: *collaborative* and *centralized.* Collaborative collaboration is a remnant of co-location; it lacks an entity dedicated to coordination. It considers all participants as equal partners, as at first seems appropriate in an ecosystem of co-producers. In this pattern, which is current practice, every team will start communicating with every other team to try and figure out the root cause for its problem and the best thing to do about it. There are, however, two fundamental drawbacks. First, because the operation and executive scope are missing, they are unlikely to make the *right* decision, as described in Fig. 1. Second, the extremely wide bandwidth of communication is both inefficient and distracting.

In the *centralized* collaboration pattern coordination is handled by one entity comprising both operation and executive. The electric design team will communicate its problem to one, and only one, entity. Because the perspectives of all three roles are now involved, the entire context is available to support an informed decision as illustrated in Fig. 1. Accordingly, individual teams will be coordinated by operations only when needed. Additional changes that are side products of the first will be managed and coordinated in the same way. The centralized pattern supports both making the best decision and the most effective way to coordinate change. It also minimizes the need for cross silo collaboration.

2.4 Communication and Coordination across Silo Disciplines

We can now pose a critical question: what information must be communicated between operation and each discipline in Fig. 2? The simplicity of the answer may seem counter intuitive: the discipline must receive all the information it requires to do its work; operations must satisfy their need to monitor progress so that they can both continue to detect unpredictable events and contain them.

The principles outlined above are used to formulate the basic encapsulation of our model as is described in the following section. The centralized pattern represents an encapsulation of coordination work between autonomous entities. Each discipline in the centralized pattern described in Fig. 3 represents an autonomous entity. The pattern encapsulates each discipline from the provider ecosystem. By doing so it simplifies the problem of coordination from the unmanageable space of an ecosystem to a simple space between two entities. This space is well

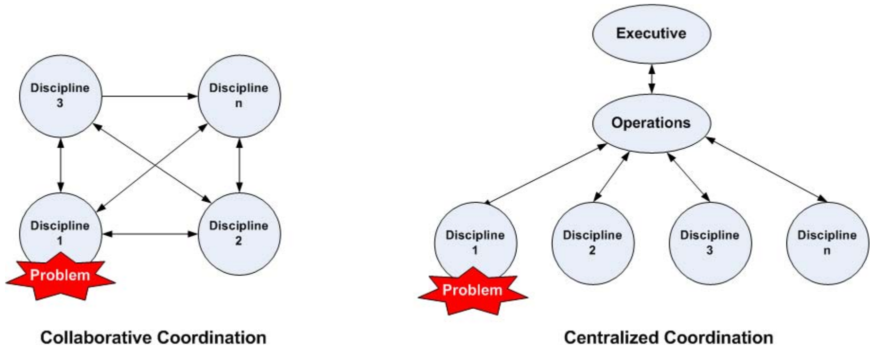

Fig. 2. Collaborative and centralized coordination pattern. Collaborative coordination taxes the communication bandwidth and because it lacks the operation and executive perspectives, is unlikely to yield a satisfactory response. The centralized pattern minimizes the communication bandwidth and lead to good decisions.

understood and can be easily managed. Furthermore, the pattern clarifies what flow of information is required between a single discipline and the coordinating entity.

Figure 3 depicts a proposed standard for Engineering Change Order (ECO) [11]. This process is critical to development and is used to manage ongoing changes. The breakdowns identified by the Aberdeen Group that were mentioned in section 1 are also related to the BPM complexity of coordinating many processes across the silo disciplines. However, as explained above, the execution of the ECO process with one discipline is well understood and can be well handled. The encapsulation provided by our model reduces the the BPM complexity to management of individual processes, such as this ECO, which is well understood and can be easily managed.

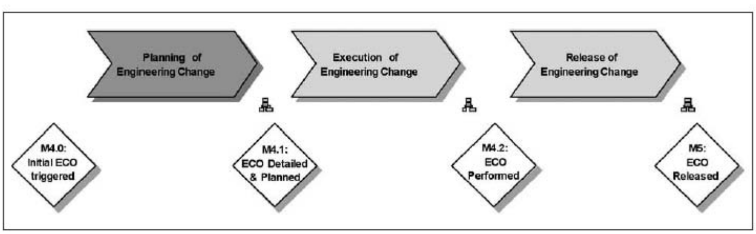

Fig. 3. This is the proposed standard for Engineering Change Order proposed by ProStep [11]. This process also illustrates the information flow between the hub and a provider. To the provider it specifies what it needs to do its work. To the hub it specifies the milestones that enable the hub to monitor progress.

3 A Model of Cross Enterprise Collaboration (CeC)

Our model of CeC work is depicted in Fig. 4. There are three central ideas: a hub that both provides the encapsulation of each organization from the ecosystem and supports the collaboration, the application of the centralized coordination pattern, and the encapsulation of domain work as a service (WaaS).

The basic components required by the hub include organizations, people, and work. In all these components we distinguish between *templates* and *instances*. Templates are designed to be reusable in a wide range of situations. They provide a standard way of doing things in a way that yields predictable results. For example, the ECO described in Figure 4 represents a process template for change management that, being defined as an industry standard, is designed for reuse across any discipline [11]. Instances, on the other hand, are designed to provide the flexibility and agility that are required at runtime. Instances are created only when needed and are then configured in the context that they are needed. This configuration is where specific information, such as requirements, defects, or designs, are specified. Configuration is the mechanism through which templates are optimally adapted to a specific run-time context. However, configuration can continue through the instance lifecycle. This enables ongoing adaptation that is critical for the containment of unpredictability. For example, in response to the problem identified by the electric design team in section 2.2, the hub may decide to modify an ECO instance being serviced by a different discipline, such as engineering. Runtime changes may add defect and design artifacts obtained from the electric team, or even add additional milestones that will verify that the new problem has been appropriately addressed.

Different hubs may modify some components to meet specific business needs. For example, additional organizations can be added to model suppliers, retailers, or customers. Therefore, the model presented here can be thought of as a meta-model that can be tailored to the specific way an enterprise operates. We next describe the necessary and sufficient components that are required by the hub

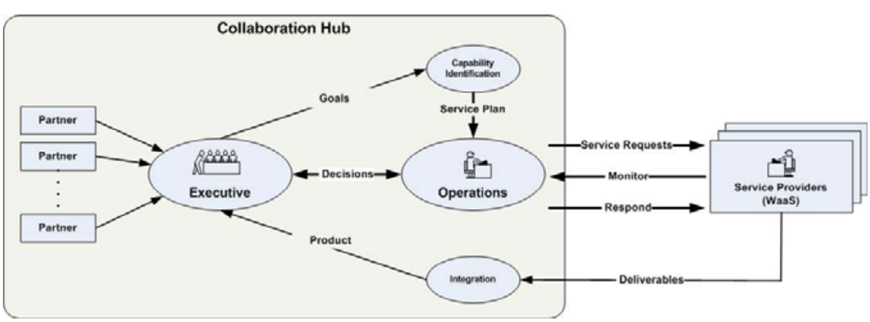

Fig. 4. The hub supports the collaboration across the ecosystem. It simplifies the problem of coordination by encapsulating each provider from the ecosystem. Depicted are the main hub teams that govern the collaboration and the flow of information between the hub and a provider organization.

and then provide an intuitive description of its operation. In the discussion that follows we explain how the model can support agility of operations across the collaboration ecosystem.

People are modeled by templates as roles, skills, and the organization type they belong to. Teams represent instances of people that together perform some common function. For example, a hub has, among others, an executive and operations team. Teams can be virtual, and the number of people in a team can be anything from one and upwards. A service provider has a *coordinator* team. Because a service provider is encapsulated, it own teams within each discipline are not required by the model.

Organizations are differentiated by role. We discuss *partner* and *provider*, but other roles exist, such as *supplier, retailer* or *customer*. An organization has two main attributes: *capabilities* and *roles*. Capabilities represent different types of work an organization can perform, and are explained below. For example, software development disciplines will have the capability to *design, build*, and *test* software. The hub itself, as it represents the virtual organization of partners, will have its own capabilities. For example, *project operation management*, *issue resolution*, or *capability identification*. In addition, each organization type specifies the roles that are required to support the collaboration. Each provider must have a coordinator role. Every hub must have an executive and operation role. Additional roles can be added to any organization.

Work is modeled by templates we call *capabilities* and instances we call *service requests*. The hub provides a fundamental encapsulation for each organization in the collaboration. The *doing* of work, therefore, is always carried out by some organization.

The prime function of capabilities and service requests is to specify the *flow of information* between the hub and a single organization. The necessary and sufficient information is (1) the information required by the organization to do the work and (2) the checks and balances required by the hub to monitor progress and detect issues. To do its work a discipline will require artifacts, such as requirements, use cases, and designs. Specification of process steps on how to do the work are usually not required as it is expected that each discipline follows its own methods and best practices. Checks and balances can include, among other, milestones, metrics, policies, and regulations. This appears quite similar to the ECO standard process shown in Figure 3. However, when a hub is set up, the collaborating partners may modify work templates to reflect their specific way of doing business. Moreover, they may agree on specific policies or regulations that will reflect their approach to governance. Thus, even on the template level, the hub provides a general mechanism for an enterprise to tailor its processes and governance.

Different capabilities represent different disciplines and their specializations. For example, each discipline may have its own capability for *requirements, design, build*, and *test*. Capabilities may be more specific, such as design of *security, GUI*, or *system*. Each capability, therefore, represents a *type* of work that is defined in terms of a discipline and skills that are required to do it. When work is needed, a capability template is instantiated into a *Service Request*, configured, and then dispatched to a provider. For the provider, configurations will specify information such as schedules, deliverables, and specific artifacts that are required to do the work. For the hub, configurations will include milestones, metrics, and policies.

The complexity of the overall work requires that different parts be carried out by different providers. A *Service Plan* is the construct that both brings together the different capabilities and coordinates the overall doing of work. Because of the provider encapsulation, a service plan is relatively simple from both process definition and management perspectives. This relative simplicity has been well documented by Malone in his theory of coordination [9] and work [10]. In particular, Malone identifies that there are only three basic types of dependencies among activities: flow, sharing, fit (aka *sequence, fork*, and *merge*). Accordingly, a service plan need only order the required capabilities according to their dependencies in sets of sequence and parallel. This is a well understood problem that can be easily managed. The service plan provides a fundamental simplification of process coordination across an ecosystem.

This simplification of process definition implies that creating a service plan is relatively straight forward and can be done quickly. Furthermore, once a plan is created, the instantiation and configuration of each capability is also straightforward and can therefore be done quickly. The time from identifying a need for work to dispatching each service plan to a provider can be relatively short. We therefore introduce the notion of *Just in Time Service Plans*, to highlight the potential for this agility [12].

3.1 Example

We will now illustrate the operation of the hub described in Fig. 4 through an example. In this scenario several companies decide to collaborate on the development of a new car. To support this collaboration they are going to instantiate a collaboration hub. Representatives from each partner join an executive and operation team. Together they define the high level business goals and strategy. They then focus on two things: how they want to govern; and how they want to manage their collaboration. The output of this process will identify two sets of capabilities: domain and hub. Domain capabilities represent different disciplines, such as software or engineering. Capabilities can be defined for *requirements, design, build*, and *test* within a domain; or even be more specific, such as design of *security, GUI*, or *system*. Each domain capability, therefore, represents a type of work which the hub will require.

Hub capabilities will include templates for frequently recurring activities, such as change management, issue resolution, and on boarding providers. For some capabilities, such as change management, industry standards may provide a starting point. The teams may decide to modify the standard so that it better reflects their particular way of doing business. For example, they may require that before any work can be assigned to a provider he must be validated. To do this validation the teams would define a new hub capability called *verify-provider*. It could test for qualifications, require specific IT configurations, or impose specific governance polices that must be carried out by each provider. The benefit is that this collaboration-specific way of doing things can now be applied generically before any work is assigned to a provider.

The second outcome is identification of additional teams that comprise the hub. In our example, four additional teams were decided upon: *operations, capability identification, integration*, and *issue resolution.* This identification reflects the partners decision to consider these three activities as core to their collaboration. The implications are that the teams will comprise members from each partner, rather than from some provider. The partners themselves will comprise the organizations that provide these capabilities to the hub as a service. Next, the partners would identify specific organizations that will service domain capabilities. This activity could be managed by a hub capability called *onboard provider*. Providers are added to the hubs list of qualified providers.

Now that the hub has been setup, it can begin to operate. The executive team would determine its immediate goals. Based on these goals, the hubs capability identification (CI) team would identify what capabilities are required to achieve these goals. These capabilities would be assembled in a *service plan.* The service plan would be passed to operations. For each capability, operations would identify the optimal provider and configure a service request (a rigorous formulation is provided in [6]). The service request would be dispatched to each provider, who would in turn, do the work. Once the work is completed, the provider would hand over its deliverables to the hubs integration team. When all deliverables from all providers are integrated into the final product, it is delivered to the partners.

So far we have described the flow of service requests and deliverables between the hub and a provider. Another critical aspect of information flow is *monitoring.* This provides the hub with an ability to satisfy itself that the work carried out by each provider is executing according to plan. Refereeing back to the example we gave in section 2.2, the hub would have detected that the electric design team may be two months overdue well in advance. As a result, the hub would automatically instantiate its *issue-resolution* capability. The configured resolution service request would identify the stakeholders from the domain team and its own operation and executive teams, collect the relevant information from the domain, operation, and executive domains, and support them in deciding what to do about as illustrated in Fig. 1. When a decision has been made, modification to ongoing work will be coordinated by the hub with each provider in accordance with the centralized coordination pattern described in Fig. 2. This coordination will be communicated between the hubs operation team and the providers coordination team. If new work is required, the hubs CI team will identify the required capabilities and assemble a service plan. Then, operations will instantiate the plan and dispatch it to the appropriate providers.

4 Discussion

The model provides two main benefits: simplification of the distributed BPM problem and a conceptual framework for a business to reason about its operations. By encapsulating each provider away from the ecosystem the BPM problem between the hub and each provider is simplified and easily managed. By encapsulating the coordination of work across the ecosystem in a hub, the complex problem of cross enterprise BPM is both bypassed and managed uniformly by a dedicated entity. The hub is a reusable meta-framework that is designed to handle the complexities of coordinating distributed work. It can be tailored to different industries, and instances will reflect the business practices of the collaborating partners. This removes the burden and complexity of needing to support coordination through the choreographies of business processes. This simplification further allows the process designer to focus on the unique business needs which the process is addressing. The design around the centralized collaboration pattern provides a reusable framework to coordinate work and contain unpredictability.

To the business the model provides a conceptual framework for it to reason about its operations. Different industries may adapt the model to their particular needs by adding, for example, customer or supplier organizations. Each business would populate the model according to its own goals, governance, policies, and best-practices. The populated model would represent that particular enterprises knowledge. This knowledge, in turn is used to drive its operations. As experience and insights are gained, they can be incorporated into the model, and immediately become operational.

Consider, for example, the root cause for the Airbus-380 failure: the German and Spanish teams used CATIA version 4, while the British and French teams used CATIA version 5 [13,4]. The collaboration failed because it did not implement a capability to onboard a new provider and check, among other things, for IT incompatibilities. Had this collaboration been supported by a hub, then the hubs *validate-provider* capability would have been used before any work was assigned to any provider. Consider, for example, the ECO in Fig. 4. Before executing any step defined in the ECO, the hub would execute its validate-provider capability. The ECO is an industry standard process. The validate-provider is a collaboration specific process. The model enables a business to tailor an industry standard process to the specific business context of the collaborating partners.

5 Conclusions

In this paper we have introduced a new model for CeC through an intelligent hub. We have shown how our model addresses the key challenges facing proper implementation of CeC: unpredictability and responsiveness. Specifically, by encapsulating each service provider and customer (or any other interacting entity) in its own space and supplying it with necessary and sufficient information, the complex BPM problem for a dynamic ecosystem is reduced to many sub-BPM problems that are known and tractable. Furthermore, our hub is designed to facilitate both the identification and the solving of two root issues of unpredictability and services evolution. The former is achieved through enhanced visibility, metrics, and traceability whereas the latter is achieved through modularity of work and service providers which are all key aspects of the hub design.

Furthermore, we have demonstrated the scalability of this framework, its ability to bypasses the complexity of cross-enterprise BPM, and its flexibility to be tailored to different industries and specific enterprise policies.

References

1. Schmidt, R.: Web services based architectures to support dynamic interorganizational business processes. In: Jeckle, M., Zhang, L.-J. (eds.) ICWS-Europe 2003. LNCS, vol. 2853, pp. 123–136. Springer, Heidelberg (2003)
2. Oppenheim, D., Ratakonda, K., Chee, Y.-M.: Enterprise Oriented Services. In: Dan, A., Gittler, F., Toumani, F. (eds.) ICSOC/ServiceWave 2009. LNCS, vol. 6275, pp. 82–95. Springer, Heidelberg (2010)
3. Grefen, P.: Towards Dynamic Interorganizational Business Process Management. In: Proceedings of the 15th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE 2006 (2006)
4. http://en.wikipedia.org/wiki/Airbus_A380
5. Aberdeen Group., http://www.aberdeen.com/Aberdeen-Library/3365/RA_Mechantronics_CJ_3365.aspx
6. Bagheri, S., Oppenheim, D.: Optimizing Cross Enterprise Collaboration using a Coordination Hub. To be published in the Proceedings of the first Global Conference on Service Research and Innovation (SRII), San Jose, USA (2011)
7. Bhattacharya, K., Caswell, N., Jumaram, S., Nigam, A., Wu, F.: Artifact-centered operational modeling: Lesons from customer engagements. IBM Systems Journal 46(4) (2007)
8. Bhattacharya, K., Hull, R., Su, J.: A Data-Centric Design Methodology for Business Processes. In: Cardoso, J., van der Aalst, W.M.P. (eds.) Handbook of Research on Business Process Modeling. Information Science Publishing, and imprint of IGI Global, Hershey, PA, USA (2009)
9. Malone, T.: What is Coordination? Paper presented at National Science Foundation Coordination Theory Workshop (1988), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.8170&rep=rep1&type=pdf
10. Malone, T.: The Future of Work. Harvard Business School Press, Boston (2004)
11. ProStep: Engineering Change Order (ECO), PSI 3-2 (Draft), Version 0.9. ProStep iVip., http://www.prostep.org/en/downloads/recommendations-standards.html
12. Hull, R.: Private communication (2009)
13. Dassault Systemes, http://en.wikipedia.org/wiki/CATIA