

Enterprise Oriented Services (Position Paper)

Daniel Oppenheim, Krishna Ratakonda, and Yi-Min Chee

IBM T.J. Watson Research Center
19 Skyline Drive, Hawthorne, NY 10532, USA
{music, ratakond, ymchee}@us.ibm.com

Abstract. We describe a service-oriented framework that supports how distributed enterprises can collaborate on doing work. Our model separates the concerns of *doing* from *managing* work. Work is modeled as a *capability* and can be provided as a *service* by some organization. A desired business outcome can then be described by its required capabilities. The framework enables dynamic composition of capabilities into *just-in-time service plans* that can be executed collaboratively by distributed organizations. A Hub is used to *manage* and coordinate the overall work. It comprises stakeholders from the collaborating organizations. The Hub's infrastructure enables them to see the big picture, detect issues early, decide on the best response, and quickly enact their decision. Executing service plans can then be modified in real-time, providing the enterprise with agility and flexibility.

Keywords: enterprise, adaptive processes, agility, human-services, work.

1 Introduction

It is widely recognized that an enterprise's survival and marketplace leadership largely depend on both the effectiveness and flexibility of its operations. As the marketplace evolves, or when unpredictable things happen, business must respond quickly. Enacting a response can be complicated, costly, and timely, especially when several organizations are affected. This is the problem addressed by this position paper: agility of business operations across distrusted organizations. We present a model and framework that can optimize the operations of collaborating enterprises through real-time modularity of work. As our focus is the overall enterprise, and our approach is to model work as chunks that can be serviced by interchangeable organizations, we term this concept *Enterprise Oriented Services*. Our approach can be viewed as a significant generalization of the "Distributed Enterprise Services" pattern [1, 2] combining Malone's Theory of Coordination [3].

There is an inherent tension between the need for standardization and predictability, that is essential for planning and marketing, and the need for flexibility and adaptation during execution, which is essential to meet the stated goals. Successful completion of complex projects depends on being able to identify problems early and then quickly enact an appropriate response. This inherent tension between *formal* and *flexible* is

clearly understood in the software development community as manifested by the *planned* and *agile* approaches [4].

Services Oriented Architecture (SOA) recognizes the need to blend formal structures with flexibility. Dynamic and adaptive processes that can “continually morph themselves to respond to environmental demands and changes without compromising operational and financial efficiencies” are not yet a reality, but have been identified as a research challenge. [5]. In our context of distributed work, the lack of flexibility is a severe limitation. An additional limitation is the paucity of support for people, and as will be explained later in this paper, especially for the role they must play when work is distributed. The addition of Human Tasks to BPEL is an attempt to address “how people can interact with business processes” but does not provide additional flexibility [6]. BPEL4PEOPLE interaction patterns, which include *4-eyes*, *escalation*, *nominations*, and *chained execution*, clearly do not address the needs of globally distributed teams that must be coordinated around a common business goal.

Recent developments provide a significant step toward bringing people and work closer to process. The *Amazon Mechanical Turk*, for example, exposes the work a person can perform as a service. This idea is expanded by the notion of *human services* in the context of Cloud Computing with the addition of composability through mashups [7]. Kern [8] initiated work on managing the quality of human-based eService. Vasko [9, 10] integrated orchestration of information derived from WS-BPEL with role-based collaboration of process participants in a web-service intended to intensify human interaction in social networks. Shcall [11, 12] expanded SOA even further by intermixing *Human Provided Service* framework with Web-Services.

Our approach separates between the *doing* and *managing* of work. We model the *doing* of any work by an organization as a *Capability*. This enables ‘chunks’ of work to be performed as a service by any qualified organization, and to a large degree, also encapsulate domain specific work. The capabilities required to meet a certain business goal can then be composed together in what we term a *Just-In-Time Service Plan*, providing modularity and composition. The framework dynamically assigns each capability within a plan to the most appropriate organization, and then provides continuous feedback on progress. This allows us to focus on the most critical issue: the management and coordination of the overall work. For this our model introduces the concept of a *Hub*. We discuss the Hub from a people and infrastructure perspective. The goal of the Hub is to bring together key stakeholders from the collaborating organizations so that they can see the “big-picture”, identify issues early, make informed decisions that balance conflicting concerns, and then enact these decisions. The infrastructure components of the Hub provide the necessary support, emphasizing dynamic composition of work as needed and runtime agility.

Next we provide three use-cases that demonstrate the problem and outline core requirements. We follow with a system overview of an enterprise and its operations that clarify important aspects that require support. We then present our model of cross-enterprise collaborative work, and end with a discussion.

2 Representative Use-Cases

The following three use-cases are from different industries and are typical of large enterprises that are globally integrated. They all boil down to this requirement: *agility*;

e.g. an ability to quickly and easily change operational procedures across an ecosystem of collaborating organization. The cases are real, but some details have been changed to protect the identity of the organizations.

2.1 Wireless Telecommunications

A leading US communications provider **A** had a strategic vision comprising two goals: increasing its speed to market, and increasing the yearly launches of new offerings from dozens to hundreds. Their business faced three fundamental challenges: intense competition, retention of existing customers, and an insufficient revenue stream. The key issue was losing competitiveness due to a lack of agility in responding to its competition. A careful analysis determined that in order to remain competitive this company must quickly introduce differentiating offerings; or simply put, reduce its time-to-market. The following goals were identified: (1) reducing the time required to enhance existing products or produce new products, (2) decreasing both development and maintenance costs, and (3) including customers and partners as co-creators of new value.

Processes governing existing products were optimized using LEAN [13, 14]. But when new products required a new process things remained problematic. Moreover, if a new product required a new capability that could only be obtained through partnership with another company, integrating **A**'s business processes with the new organization remained a stumbling block. Our model addresses this problem by (1) modeling *Capabilities* as services that can be executed interchangeably by any qualified organization, and (2) enabling new offerings to be modeled as *Just-In-Time Service Plans* that can operate across organizational boundaries. The need to include customers as co-creators is addressed by the Hub's infrastructure that provides a "plug-and-play" mechanism for both partners and providers.

2.2 Financial Sector

A leading financial corporation **B** has transformed over the past two decades from a Multi-National to a Globally Integrated Enterprise (GIE). It has presence in more than 50 countries. Over the years it has carefully placed clones of its business units in countries it identified as strategic. Each unit was given the autonomy to leverage local opportunities, and therefore adapted to local practices, policies, and regulations. Whereas each unit is doing similar work, in reality their processes vary considerably. As a result **B** is now experiencing pain in monitoring and coordinating its global operations. Moreover, finding out the true financial state of the corporation has become so complex that it was only done once a quarter and required intensive manual labor by highly skilled people. After a careful analysis **B** concluded that it wanted a drastically reduced set of simplified business processes that will be used uniformly by every business unit, yet still allow individual units to adapt locally. It also wanted the ability to provide its executives with a "real-time" view of its finances.

The two fundamental issues challenging **B** are (1) a standard way to manage similar work, yet allowing local variations for handling geo-specific policies and regulations; and (2) overall operational visibility and transparency. Our model of work as a *Capability* can provide the overall standardization required by **B**. Encapsulation of the

doing of work as a *service* permits individual organizations to make local changes within their internal procedures. As the overall operations would be managed utilizing a *Hub*, operational visibility and transparency across the global units would be supported on a primitive level.

2.3 Automotive Industry

A large company **C** in the automotive industry has its own business unit for developing the IT required by all its product lines. Over the last decade this unit transformed from a centralized organization to five centers that are distributed world-wide. The initial motivation for this was to leverage labor arbitrage. However, following the Wipro Factory Model [15], each location also specialized in a different aspect of the Software Development Lifecycle. Accordingly, each phase in the *Design, Build, Test, Deploy* and *Manage* development lifecycle is now executed by a different global team.

The main challenge facing **C** is the ability to effectively manage its IT development process. As it was originally designed for a centralized and co-located organization, it lacks the ability to coordinate, monitor, and govern new projects across its new distributed organizations. Our model of work as *Capabilities*, and *Hub* as a cross-enterprise operations center address these new needs.

3 The New Enterprise Eco-system

Globalization, internet, and technology are having a profound impact on all aspects of the enterprise, its organization, and the way in which people perform work. Freidman [16] describes the emergence of “Globalization 3.0” in 2000; ten forces and a triple convergence are shaping every aspect of business, politics, and society. Verna describes how businesses are “evolving into the networked patterns of living systems” and defines the organization as “a complex adaptive social system where people systematically cooperate to achieve a common purpose” [17, p. 14]. Malone describes how the low costs of communications are transforming society, businesses, and work and notes a move towards decentralization that empowers people and allows them to participate in decisions that matter to them [3p. 5]. People have always been at the center of business operations, but globalization is empowering people world-wide in radically new ways.

The enterprise is a complex system with many facets, including strategy, operations, organization, people, partners, co-producers, or service providers. The enterprise eco-system is represented by many different stakeholders that can each have different concerns that may even conflict with those of others: risk, quality, governance, intellectual property, tooling, methods, policies, or regulations. It is only by taking a broad systemic view that both recognizes and supports this seeming complexity of facets and concerns that a simple model that will support the broader operations of an enterprise can be devised.

3.1 A System View of the Enterprise

Figure 1 depicts a simplified system view of an Enterprise. The top of the diagram outlines its global eco-system of the marketplace, suppliers and retailers, customers,

partners, etc. This dynamic value-networks is constantly changing, evermore so with the accelerators of “Globalization 3.0” [16]. Each member brings with him capabilities that would otherwise not be available to the enterprise and all members collaborate to achieve a greater value.

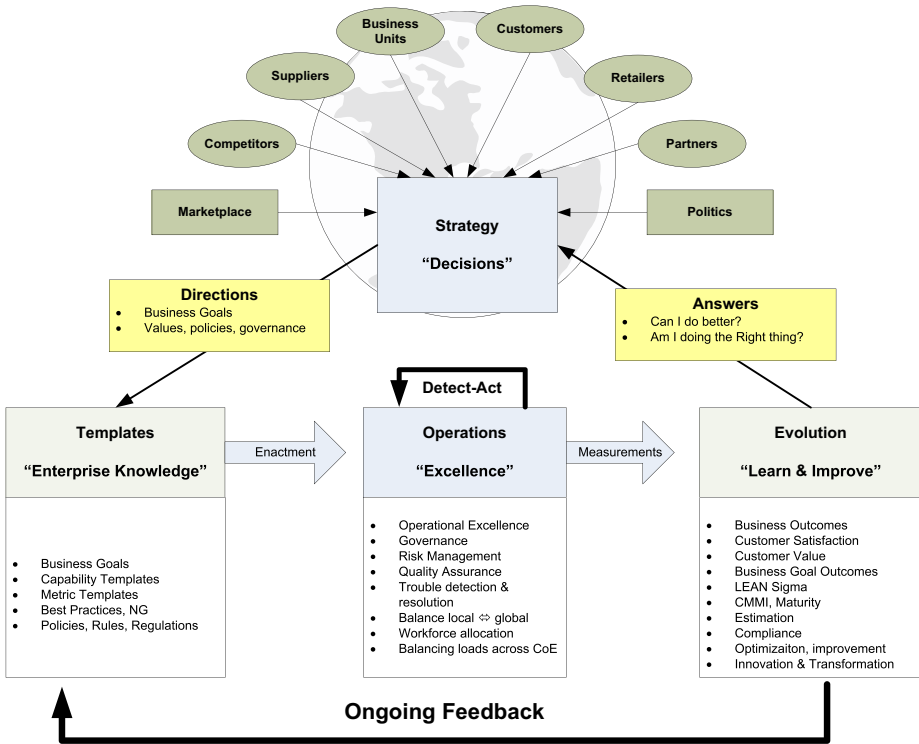


Fig. 1. An Enterprise system view depicting its dynamic eco-system, relationship between Strategy, Operations, Knowledge, and Evolution, ongoing feedback loops, and the role of measurements and governance

Two important aspects are *Strategy* and *Operations* and the way in which they interact within the enterprise system. Strategy is primarily concerned with winning the market and is responsible for deciding what would be the “best” goals for the business. Given a goal, a careful analysis can determine the capabilities required to accomplish it [18]. E2E business processes are then designed to optimally control the flow of work between the different business units and organizations in the value-network. In the current state-of-the-art, specialized business applications, such as SAP or ORACLE, are used to automate work-flow. Supporting flow in such applications is costly, timely, and lacks flexibility once complete. Organizations can be therefore tempted to resist innovation and transformation but that can have severe consequences [20, 21]. Our framework to be described encapsulates such applications from process and flow, simplifies process design, and significantly shortens enactment time.

Operations are concerned primarily with getting things done well. Excellence is usually associated with the capability of doing work in a way that is consistent, repeatable, and therefore predictable. Operations are usually measured in terms of time, effort, and quality. Quality is often measured by a combination of customer satisfaction and post-delivery defects. As already noted, there is a fundamental tension between the desire for standard-repeatable-predictable operations and the fluid, often unpredictable nature of work. Our model addresses these concerns.

Two additional important dimensions are *knowledge* and *evolution*. The enterprise knowledge is its differentiating “secret sauce.” It comprises its best-practices, business processes, capability ‘how-to’ guides, measurement and monitoring strategies, and so on. Knowledge is important not only to effective operations but also to informed strategic decision-making. Harvesting new knowledge to update the knowledge base is highly desirable. The ability to act on new knowledge is a starting point for ongoing evolution.

3.2 Two Critical Feedback Loops

Two critical feedback loops can be seen in Fig. 1 that supports operations, strategy, knowledge, and evolution. As Strategy makes decisions regarding business goals and directions, it must also monitor their outcomes so it can answer questions from three fundamental perspectives:

1. *Operations*: am I functioning in the most effective way to meet my goals?
2. *Strategy*: Am I achieving my stated goals? Are my goals still the right goals or should I replace them?
3. *Innovation*: what can I do to improve my operations or strategy?

The data required to answer these questions is a combination of metrics collected during operations, its own knowledge base, and information from the marketplace. The ability to answer these questions is done through specialized analytics that is carried out by the Evolution team. Answers to these questions trigger innovation and transformation.

Operations have their own special feedback loop to enable what we call on-demand *detect-act* activities. It denotes the ability to detect an issue at the earliest possible time and take immediate corrective action (see discussion in the context of supply-chain [22]). In the context of people doing work this is critical for operational excellence, quality, coordination of work, and global governance. Our model inherently supports all of the loops. Runtime flexibility enables deploying new sensors to address new and unforeseen situations, alongside the triggering of their immediate, well governed resolution. Runtime flexibility then enables modifying both plans and services as required. Analytics deal with post-work analysis, the second feedback loop, which provides answers to strategy that can lead to ongoing evolution.

4 A Model of Work

This section defines our model of work leading to its encapsulation as a service. We factor work using an object-oriented methodology and take an artifact-centric approach to define the core model elements [1, 2, 14, 23]. In this way our model is

meaningful to stakeholders in the business domain yet also forms the underpinning to our IT framework.

4.1 Definitions

For the sake of clarity and brevity our definitions are applicable only within the context of this paper.

- *Business Goal* defines some value that the enterprise wants to create. One or more capabilities may be required to obtain this value.
- *Capability*: the ability to perform actions that generate an outcome. As applied to human capital, capability is the product of expertise and capacity. (Adapted from Wikipedia). Our model associates a given capability with only one entity that will service it: a work-hub.
- *Compound Capability*: A capability can be composed of child capabilities and their relationships which can be represented in a directed acyclic graph (DAG).
- *Work*: human effort applied to realize a capability by performing the actions that generate its outcome (adapted from Wikipedia).
- *Capability Template*. A business artifact that specifies the attributes required for (1) providing the people who do the domain work with all they need to execute, and (2) providing the work-hub with all that it needs to manage, monitor, coordinate, and govern the work. Templates are static documents that represent a “best practice” of how to do work.
- *Enterprise Service, or Service*. A service is an enacted Capability Template that is appropriately configured, in the context of its execution, with the correct attribute values (data), people (team), and lifecycle (process). A service provides a team with everything required to *do* its work, and a work-hub with everything it requires to *manage* ongoing work, including monitoring and governance.
- *Service plan, or just-in-time Service Plan*. A service plan is a structure that contains one or more enacted services and their relationships that together service some higher level capability. Service plans are lightweight objects that can be easily assembled by reusing available Capability Templates. The structure of a plan can be modified at runtime, as can any aspect of a service.
- *Team*: a virtual entity comprising one or more people that *do* the work required to realize a capability. A team belongs to one organization. Each role definition specifies responsibilities, accountability, and skill level.
- *Work Hub*: A virtual organizational unit that is responsible for *managing*, coordinating, and governing the work around a service. The hub manifests our separation of concerns between *doing* and *managing*. Simple service requests could be implemented by one person. At the other extreme, complex requests may represent enterprise-level compound capabilities. A hub could then internally breakup this request into a service plan in which different capabilities could be managed by internal teams, other hubs, or combinations thereof.
- *Service Flow Engine*: the IT infrastructure that supports and automates all the above.

4.2 Artifact Centric Objects

We make a core distinction between templates and instances. Templates embody best-practices and are an important component of the enterprise's knowledge. They are created by experts, intended for broad reuse, and provide a standardized start-point for work. Templates can be continuously modified and improved with experience. Instances are managed by the Process Flow Engine. They are designed to be flexible and easily modifiable at runtime. When a Template is instantiated, its structure, attributes, and behaviors are copied into the instance. Then, specific values can be configured in the instance, such as people, dates, artifacts, policies, teams, governance, or metrics. Table 1 lists several of the more important objects.

Table 1. Templates and Instances

Template	Instance
Capability Template	Service
Role	Assigned Role
Team Template	Team
Work Hub Template	Work Hub
Metric Template	Metric
Compound Capability Template	Service Plan

The *Capability Template* and its *Service* counterpart are intended to provide everything that is required to both *do* and *manage* work. They have many attributes that are grouped to address different facets of the problem:

- *Service Composability.* The most rudimentary requirement for composability is a clear definition of *inputs* and *outputs*. In the context of people and work these are usually documents in the form of work-products, but they could be anything. Inputs specify everything that is needed by people to do their work, such as requirements, designs, or specifications. Policies, rules, and regulations are another category of inputs. Normative Guidance, examples, manuals, and anything that can help the person be more effective are a third category of inputs; there can be others. Output specifies what must be the outcome of the work; this is the value derived from this capability. The output might be a new work-product, document, or object, or it may be a coherent set of updates to existing work-products, documents, or objects; or any combination thereof.
- *Quality of Service.* Several complementing strategies are used overall to ensure operational excellence. In the context of a capability template quality of service is addressed by the specification of everything that needs to be monitored to ensure optimal operations, alongside the specification of governance when things are not. This would primarily include a list of metrics and related baselines against which they will be compared at runtime. There is infrastructure to enable the displaying of dashboards.
- *Governance* is critical when people do work, and is therefore built in as a primitive mechanism that is both systematic and dynamic. This section specifies one or more governance teams that will be assembled dynamically to

resolve any issue detected at runtime; different types of anticipated issues may require a different team composition. In addition to specifying roles, responsibilities, and accountability, a governance specification will outline the scope of the stakeholders required from the global value-network of collaborators.

- *Late service binding* covers interoperability of services. There may be more than one work-hub that can service a given capability. The attributes in this category provide information on required skills, capacity, required CMMI certification level, specific regulations, and so on. This enables the *Just-In-Time Service Plan* to dynamically allocate the most appropriate Work-Hub at runtime.

The Work-Hub template and instance represent everything that is required to *manage* work. In this respect, all work-hubs represent similar capabilities. As *managing* is different from *doing*, and as it is a capability that is required whenever any work takes place, it has earned its own representation. Different templates can represent needs of different business domains as well as needs of varying complexities of similar work. Managing the work of an entire enterprise is very different from managing work done by a single person. The following groups of attributes are worth noting:

- *Discovery*. Every hub must make available the list of capability templates it can service. In addition, the Hub publishes information regarding available roles, skills, and capacities. This is not unlike the idea of a Universal Description Discovery and Integration (UDDI).
- *Private Capabilities and Teams*. There are many activities that routinely take place that relate to *managing* work but do not contribute to the *doing* of work. Governance around issue resolution is probably common to all work. In the domain of software development, for example, one can name project-management, change-requests and defect-management. Different domains may have similar or additional activities. The concept is that by factoring out these activities from the *doing* of work, such activities can be handled by all hubs uniformly. This separation of concerns between *do* and *manage* makes it easier to evolve one without affecting the other. It also provides consistency and uniformity where it is needed without compromising the ability to be flexible and adaptive.
- *Live service plans*. Hub instances may manage more than one live Service Plans. After all, this is what the Hub is intended for.

4.3 An Operational Model of Work

The literature around Globally Distributed Software Development generally considers inter-team collaboration to be a key pain point [24, 25]. Our observation of co-located teams that must collaborate with globally distributed teams led us to different conclusions. We observed that teams enjoy working together in co-location and view frequent interactions with other teams as an annoying distraction. These realizations lead to our separation between *doing* and *managing* work. This approach minimizes the need to interact with other teams and thus maximizes the benefits of co-location. In honor of the memory of Richard Feynman we name this principle the *conservation of co-location*.

Another distinct advantage of this factorization is that we are conceptually moving from a procedural to a declarative model: we describe *what* should be accomplished (the capability) rather than *how* to accomplish it (process steps). This simplifies the composition of a *Service Plan* to the degree that it can be done *just-in-time*. It also provides runtime flexibility, especially the ability to replace providers dynamically.

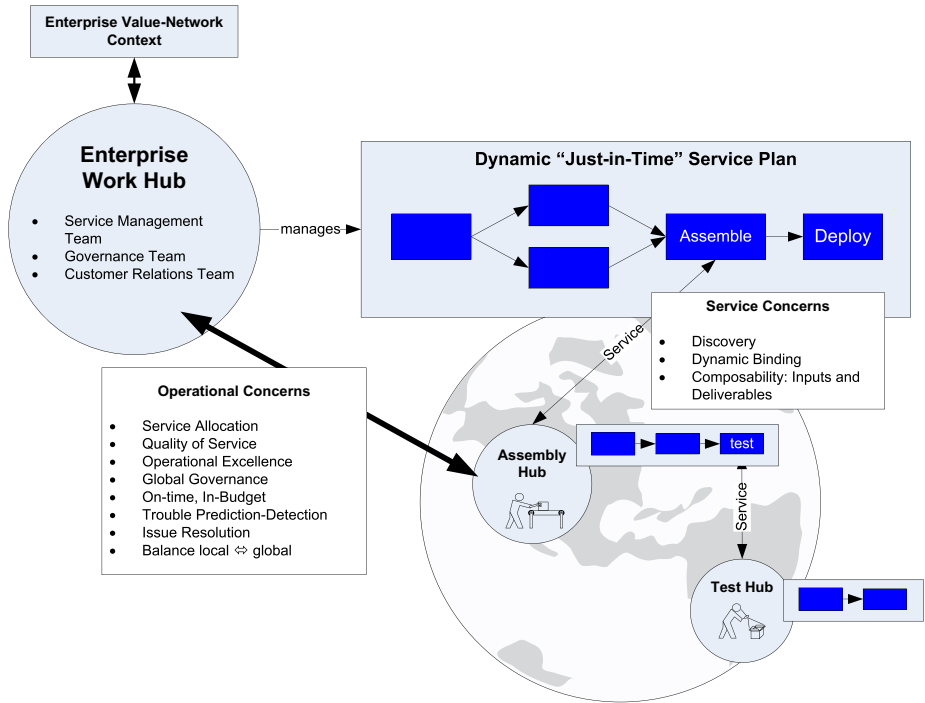


Fig. 2. The *manage-do* model of work

Figure 2 depicts our *manage-do* model which is both dynamic and recursive. The enterprise work-hub represents the highest level of the enterprise, managing the end-to-end work around some key business goal. For simplicity we have depicted a small set of representative hub-teams that deal with managing operations, governances, and the customer. The required capabilities are specified in a dynamic just-in-time service plan that is executed by the framework described in the following section. The capability *Assemble* is dynamically assigned to some Assembly Hub. In order to service the request this assembly hub uses the same model to manage its internal work. It dynamically creates its own service plan which comprises three capabilities. The first two capabilities may be worked upon by the hub's internal work-teams. However, for whatever reason, the third capability, *Test*, is dynamically assigned to a different hub. The Test Hub may be anywhere in the world, and belong to any organization.

The work-teams in each Hub need only be concerned about the *doing* of work specified by the *Service*, primarily with the *inputs* and *outputs*. Other service-oriented concerns include *discovery* and *dynamic binding*, but these are managed by the parent

hub, not the work-team. Hub-teams are primarily concerned with operations and Quality of Service. They require frequent coordination between their child work-teams and/or child or parent hubs. For example, any problem relating to the execution of the *Test* capability will be detected and handled by the *Assembly-Hub* without involving any of the work-teams unless some coordination is required. However, if the problem is serious enough, the *Enterprise-Hub* may be brought in, so that stakeholders from the enterprise value-network context, such as a customer CxO, can help make the right decisions. In this way the hub hierarchy also defines a management and governance context.

4.4 IT Framework Architecture

Figure 3 depicts a schematic architecture of an enterprise IT framework which supports globally distributed work in a service-oriented approach. At its core is a *Service Flow Engine* that can manage the instantiation of services, dynamic composition within Service Plans, flow of artifacts between globally distributed teams, and ongoing sensing for issues. And should an issue be detected the Service Flow Engine will, immediately initiate their resolution. The flow engine provides the flexibility and agility to modify any aspect of process and work.

There are four important system components. The *Templates* component supports the definition, modification, and persistence, of process templates. It is an important aspect of the Enterprise Knowledge. *Runtime* deals with template instantiation and configuration, service flow, and any related automation. *Health* assists with quality and operational excellence, primarily through a dynamic and flexible framework for sensing and alerting. *Analytics* includes a rich suite of data mining, pattern matching, prediction, and other algorithms that provide a key input to strategy, innovation and transformation. Persistence for these components is provided by several virtual data-bases. Runtime persistence provides extensive traceability such as ‘*who did what, when, and why*’.

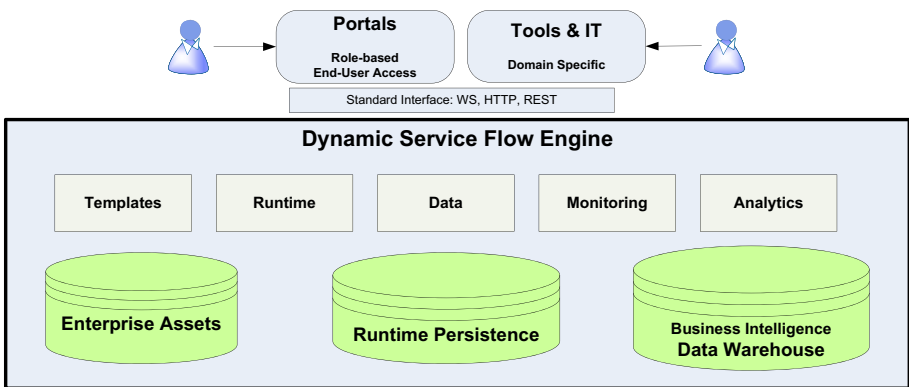


Fig. 3. Schematic architecture of enabling IT framework

Access to the framework is enabled only through carefully defined interfaces; and information is guarded by a data-centric security layer. Teams and practitioners access the framework through a portal. Access is role based and information presented to the user is further filtered by the individual person and his current activity. Tools used by users to do their work can also interact, and in some cases integrate, with the framework through external interfaces.

5 Discussion

We have presented a model of work within a service oriented framework that supports the operation of a globally distributed enterprise. We have demonstrated how SOA concepts can improve the way in which an organization can manage people doing work across geographical and organizational boundaries. Compared to the current state-of-the art, our *Just-In-Time Service Plan* provides considerable run-time flexibility alongside the agility to react through ongoing cycles of strategy & transformation. Referring back to the use cases presented in section 2 should make this more clear.

Enterprise **A**'s first pain-point was reducing "time to market" of enhancements or new products. There are two inhibitors: process modification, and then process enactment. Process modification can be achieved by either recombining existing Capability Templates into new Service-Plans, or by creating new Capability Templates that provide new capabilities. In either case, enactment is almost instantaneous when utilizing the Service Flow Engine. The second pain point of decreasing development cost is also addressed by the Capability Templates and Service Plans, and especially by their inherent ability to monitor and measure any aspect of operations in such a manner that process improvement and optimization can take place from one enactment to the other. **A**'s third requirement to include customers and/or partners as co-creators is the hardest to implement under current practice. However, as any third party who can service a Capability-Template can be included in a Service Plan, this becomes a non-issue.

B's first concern is to have fewer processes that are standardized yet remain flexible. All **B** needs to do is to decide upon one Capability Template and have that used by all its business-units. Enactment will not be a problem. **B**'s second concern is increasing visibility and transparency. This is managed by our framework on a primitive level, so it is both systemic and dynamic. All **B** needs to do is to specify which "sensors" should be deployed in each Capability Template's Quality of Service (QoS) section. Another advantage of our approach should be noted. The standard approach for monitoring and dashboards is to create a Business Intelligence (BI) Data Warehouse (DW) into which business units are required to replicate data. The limitation of this approach is a lack of flexibility in changing the DW schema when new concerns arise. Our framework provides the runtime flexibility of being able to add sensors that can address multiple stakeholder concerns, even if these concerns conflict. Thus, our runtime persistence model complements the BI-DW and can provide increased visibility that is flexible and can be fine-tuned during each enactment.

Enterprise **C**'s primary concern is coordinating its end-to-end processes that were once co-located but now span its globally distributed business-units. The Service Plan replaces the end-to-end process, and each business unit can be managed and coordinated effectively through the Capability Template's Service instantiations. This separation of concerns between the *doing* and the *managing* of work effectively supports issues relating to globalization and collaboration.

However, this is but a first step in what we believe is a promising direction. There are many issues that need to be flushed out. There are at least two critical areas we feel should be explored in depth. The first is the relationship of our model and framework to Business Process Management (BPM). The Service-Plan is similar in concept to a business processes. However, it is simpler to define, and is dynamic and flexible when enacted. It unifies process with data; supports monitoring, measurements, and governance; focuses on business capabilities and their outcomes; and is deeply recursive in structure. When instantiated, plans are both configured and customized to the specific context of the enactment. In principle, one could easily create a new Service Plan on the fly to address similar needs. This might enable a new approach to business process management and evolution.

An additional aspect which we think merits closer investigation is relationship of the framework to the new paradigms of Cloud Computing. Cloud could be an ideal delivery platform for Enterprise Oriented Services, where “everything is a service.” Cloud virtualization of applications and IT could further assist in providing application and IT capabilities that are specific to a business domain, yet can be factored away from the Service Plans and Capability Templates. This could provide a separation of concerns between the general “how-to”—Templates—and the specific “enactment”—Instances and IT.

References

1. Bhattacharya, K., Caswell, N., Jumaram, S., Nigam, A., Wu., F.: Artifact-centered operational modeling: Lesons from customer engagements. *IBM Systems Journal* 46(4) (2007)
2. Bhattacharya, K., Hull, R., Su, J.: A Data-Centric Design Methodology for Business Processes. In: Cardoso, J. (SAP), van der Aalst, W.M.P. (Eindhoven Univ.) (eds.) *The Handbook of Research on Business Process Modeling*. Information Science Publishing, and imprint of IGI Global, Hershey (2009)
3. Malone, T.W.: *The Future of Work*. Harvard Business School Press, Boston (2004)
4. Boehm, B., Turener, R.: *Balancing Agility and Discipline, A guide for the Perplexed*. Addison-Wesley, Reading (2004)
5. Papazoglou, M.P., Traverso, P., Dustdar, S., Leymann, F.: *Service-Oriented Computing: State of the Art and Research Challenges*. *IEEE Computer*, 64–71 (November 2007)
6. WS_BPEL Extension for People – BPEL4People. A joint white paper by IBM and SAP (2005),
<http://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/cfab6fdd-0501-0010-bc82-f5c2414080ed>
7. Tai, S.: *Cloud Service Engineering*. Public presentation made at IBM Research, Hawthorne NY, May 27 (2009)
8. Kern, R., Sirpins, C., Agarwal, S.: *Managing Quality of Human-Based eServices*. In: *IC-SOC 2008 International Workshops*, Sydney, Australia (2008)
9. Vasko, M., Oberortner, E., Dustdar, S.: *Collaborative Modeling of Web Applications for Various Stakeholders*. In: *Proceedings of the 9th International Conference on Web Engineering (ICWE)*, San Sebastian, Spain (2009)
10. Vasko, M., Dustdar, S.: *Introducing Collaborative Service Mashup Design*. In: *Proceedings of the 9th International Conference on Web Engineering (ICWE)*, San Sebastian, Spain (2009)

11. Schall, D., Troung, H., Dustdar, S.: Unifying Human and Software Services in Web-Scale Collaborations. *IEEE Internet Computing* 12(3), 62–68 (2008)
12. Schall, D.: Human Interaction in Mixed Systems – Architecture, Protocols, and Algorithms. Ph.D. Dissertation. Technical University of Vienna (2009)
13. George, M.: *Lean Six Sigma for Service*. McGraw-Hill, New York (2003)
14. Fleming, J.H., Asplund, J.: *Human Sigma: Managing the Employee-Customer Encounter*. Gallup Press (2007)
15. Upton, D.M., Fuller, V.A.: *Wipro Technologies: The Factory Model*. Harvard Business School Case No. 9-606-021
16. Friedman, T.: *The World is Flat, a Brief History of the Twenty-First Century*. Farrar, Straus and Giroux, New York (2005)
17. Verna, A.: *The Future of Knowledge: Increasing Prosperity through Value Networks*. Butterworth-Heinemann, Butterworths (2003)
18. MacDavid, D. W.: A standard for business architecture description. *IBM Systems Journal* 38(1) (1999)
19. Mann, C.C.: Beyond Detroit: On the Road to Recovery, Let the Little Guys Drive. *Wired Magazine* (June 2009)
20. Christensen, C.M.: *The Innovator’s Dilemma: The Revolutionary Book that Will Change the Way You Do Business*. Collins Business Essentials (2003)
21. Bliss, G.: *Principles of Modern Manufacturing and Supply Chain Management*
22. Nigam, A., Caswell, N.S.: Business Artifacts: An Approach to Operational Specification. *IBM Systems Journal* 42, 3 (2003)
23. Herbsleb, J.D., Moitra, D.: Global Software Development. *IEEE Software* (March-April 2001)
24. Cataldo, M., Herbsleb, J.D.: Communication patterns in geographically distributed software development and engineers contributions to the development effort. In: *CHASE 2008: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering*, pp. 25–28. ACM, New York (2008)