Towards Autonomic Business Activity Management

Jun-Jang (JJ) Jeng and Henry Chang

IBM T.J. Watson Research Center
Yorktown Heights, New York

Business activities can be highly complicated and dynamic. This complexity lies in the number of activities and organization, and many business relationships between the things that make up a business activity and its business environment. The dynamism of business activities comes from the needs of continuously measuring, monitoring, controlling and improving business activities. The systems providing this type of functions and services are coined as Business Activity Management (BAM) systems. A BAM system enables an enterprise to respond to emerging problems at right time. Right time does not equal to real time though, in many cases, real-timeliness is one of the requirements. Traditional techniques for building BAM systems are proven to be inadequate in dealing with the aforementioned complexity. It is not uncommon for some minor perturbation in some remote corner of the system to have unforeseen, and at times catastrophic, global repercussions. In addition to being fragile, many situations arising from the extreme dynamism of these systems require manual intervention to keep distributed applications functioning correctly. In our opinion, BAM systems should possess the characteristics of confronting the high degree of complexity and enabling the construction of robust, configurable, scalable, self-organizing and self-managing distributed systems.

A typical electronics value chain consists of retailers, manufacturers and parts/capacity suppliers. These value chain partners collaborate in the context of business operations to deal effectively with dynamic changes in the marketplace. These collaborative relationships enable synchronization of the orders, production schedule, and parts supply plan. Traditional workflow management can help automate the business activities that make up this scenario. The state-of-the-art involves modeling the business processes and executing these models in an appropriate workflow engine. The behavior of the resulting system is static in the sense that it cannot accommodate any business situations not explicitly modeled at build-time. Therefore, a workflow approach tend to create an infrastructure that will not be capable of handling business related deviations or exceptions or evolutionary changes.

We propose using an adaptive entity named *BABot* (Business Activity Bot) as the cornerstone of the BAM systems. Since the relationships among business activities and entities are dynamic, the bindings between BABots at run time are dynamic as well. A BABot consists of seven perspectives. The *context* perspective describes the contextual information for the existence and behavior of a BABot. Being aware of business contexts is essential for driving effective development of BABots. The *intent* perspective describes the goal of a BABot and relates the goal to other goals, capa-

bilities, and values. The *value* perspective is the value-exchange view of the concerned business activities. This uses the monitoring process heavily to sample and analyze the operations of target business activities. The *capability* perspective specifies and links what a BABot can do, from the strategy level to operation, to execution level, and to the resource level. The *constraint* perspective regulates the logical relationships among values, capabilities, intents, and contexts. Constraints impose a set of configurable commitments and rules upon designated BABots. The *resource* perspective specifies the resources governed by the enclosing BABot. Resources can be shared among multiple BABots. The *processes* perspective describes how BABots cooperate with one another and harness passive resources in order to execute capabilities and to desired outcomes. Events trigger the commencement of processes. Five processes are identified in BABot-based BAM systems, i.e., metric calculation, situation detection, situation analysis, decision making, and decision enforcement.

Consider the supply chain example. A conventional management system for supply chain often consists of two layers: a planning layer and an execution layer, where the planning layer allows business condition to be periodically analyzed in order to make adjustment of work scheduling, supplier contract, or crucial resource allocations. The execution layer is the daily operational process to carry out business functions by following the preset rules and policies set up by the planning layer. Our approach inserts a reactive layer between the planning and execution. Since the planning layer has to be done periodically to avoid thrashing and destabilizing the execution system, there will be process exceptions, special situations not responded well in the execution layer. To provide real-time adjustment, the reactive management layer would predict, localize and classify the situation. If the situation warrants a big adjustment of planning, an alert would be sent to the planning layer to ask for urgent rep-planning, such as the disruption of goods supply during the event of war or market crash. The following diagram shows the sense-and-response support of a supply chain through the BAM system. If the situation is repairable, the BAM system may resort to automatic analytic algorithms to assess the damage and trigger an appropriate autonomic recovery process.

Autonomic BAM system represents a natural evolution of the policy-based real-time monitoring and reactive recovery functions of a complex system. A percentage of the detected situations may be amendable for self-healing, self-management, and self-optimization with minimal human interventions. Others may definitely need human involvement. Such being the case, the mathematical modeling techniques required for an autonomic business process is not far away from that needed for an IT component such as storage subsystem. In the BAM domain, we believe that the autonomic functions will emerge by adopting better prediction and optimization algorithms for specific business domains, such as inventory replenishment in supply chain, risk containment in finance. We are developing BABot-based BAM system as the endeavor towards an enabling infrastructure for self-managing BAM systems. We will investigate cloning, versioning, and mobility of BABots with an eye towards non-functional requirements such as performance, availability, fail-over, and migration that are critical to the success of BABot in practice. This infrastructure is being applied to real-world scenarios including supply chain, logistics, finance, insurance, and manufacturing.