Activity-Based OO Business Modeling and Control

Like managers of any other business process, software developers seek ways to optimize their “production” process. Until recently, when developers shifted their focus from the product itself to the method used to produce it, they lacked a way to describe software development processes, let alone control them. Now, many organizations recognize the importance of process, which is—when you think of it—one of the most reusable “components” in many software companies. But just how do you go about describing software development in business process terms? Several attempts have been made to create reusable processes. Although level 3 of the CMM prescribes to have some sort of process reuse in place, effective means for process reuse have not yet been formalized.

Business Modeling for Control

In the 1980s and early 1990s business reengineering was not only very popular, but looked to be the only alternative to stagnation. But reengineering is costly and disruptive. For this reason, most recent process models introduced in North America and Europe tend to emphasize a different view: process change. For instance, the ability to change a process is essential to achieving the highest level of CMM certification. Now most business professionals view continuous process improvement as just another business activity.

Our team’s method, activity-based object-oriented business modeling and control, follows the software process engineering track. We based ABOBM C on proven software development models that have the representational power to describe a software company. Software company operations rely intrinsically on activity-based costing; the widely known and very effective Personal Software Process (Watts Humphrey, Introduction to the Personal Software Process, Addison-Wesley, Reading, Mass., 1996) can be viewed as a tailoring of ABC to individual developers to assess, improve, and homogenize their work. Finally, the pioneering work of Ivar Jacobson, Maria Ericsson, and Agneta Jacobson (The Object Advantage: Business Process Reengineering with Object Technology, Addison-Wesley, Reading, Mass., 1995) has resulted in the application of object orientation to business modeling.

In ABOBMC, we offer a user-selectable level of detail for the process description, and impose no limit on the amount of decomposition. In this way, you can adopt trade-off strategies dictated by the market and not by the method’s clumsiness. Conversely, ABOBMC retains enough rigor to be used in statistical analyses and quality control.

Change management and control

Desirable as it is to introduce process improvement, companies are reluctant to adopt such strategies, primarily because they cannot estimate the benefits of small changes. The introduction of measurement frameworks such as the balanced IT scorecards (ESI, European Software Institute, http://www.esi.es, 1997) has improved this scenario, but we still need a coherent change descrip-
tion system that can provide a basis for estimating the impact of change.

Moreover, organizations find change desirable only when it is kept under control. Although many process models encourage instituting change management systems, they rarely go into the details of change control. A B-OOBMC’s core supports change control management directly by monitoring the individual activities, thus coping with the needs of nimble, smaller companies while maintaining the abstraction power needed to describe larger, more stable ones.

Activity-based object-oriented business modeling and control

A B-OOBMC relies on a model that describes the software process. We base this model on four basic entities: activities, people, infrastructures, and roles. Activities are further divided into three classes: interface, control, and execution, as shown in Figure 1. The conceptual relations between the entities are shown in Figure 2.

Next, we group the main entities into diagrams, with each diagram serving as a tool to explore a particular perspective of the model. Each diagram groups only some of the entities, linking them according to a specific set of relations. For example, roles relate to processes by activity profiles, which detail how much effort people playing a role spend on a process. The complete model includes diagrams and free-form text. The text comes from an initial system description, based on use cases described in personnel interviews.

We build the complete model using an iterative cycle with the following structure, diagrammed in Figure 3:

- Identify use cases. The first diagram in the model, the use cases capture people’s usual activities, as well as any heuristics practices for coping with exceptional situations. Ideally, this phase correctly identifies all the entities involved.
- Identify people. This phase records all the people involved in a set of activities. At first, it may be unclear in particular cases if a single person is part of a process. However, the best strategy at this point is to include the person. This phase also involves consulting human resources records to extract each person’s skills.
- Construct people-interaction diagrams. Modelers can now review the use-case interviews with a clearer idea of the people involved. They use this information to produce a series of new diagrams that depict the flow of activities in terms of message interactions among people.
- Identify roles. A structure begins to emerge from the previous diagram. Modelers now review the diagram and use cases to assign roles to each person, then try to generalize these roles into the most generic role necessary to perform a certain task. Modelers also refer to the company’s organizational chart to complete this phase.
- Identify activities. Modelers now subdivide each task into activities, grouping some elementary activities into more complex activities, also called processes. Again, modelers refer to the use cases and try to find a common set of activities performed across a range of tasks. For example, people perform the activity “consult documentation” in hardware and software design, testing, accounting, and many other tasks.
- Construct activities-roles-people snapshots. At this point, modelers can summarize the complete process being modeled with an A R P snapshot—essentially a very high level diagram that shows the process’s general structure. Particularly easy to read, this snapshot lets every participant in the process review and give feedback.
- Construct activities-interaction diagrams (A I D s). Now that the modelers clearly understand people and roles, they use A I D s to seek a more detailed understanding of activities and message exchange.
- Construct activities-roles-people diagrams (A R P D s). Modelers build this final part of the model to acquire an analytical representation of each activity in terms of the people and roles involved.
Object orientation plays a major role in the diagrams, since it allows aggregations of entities—for example, processes as aggregations of activities. It also allows families of similar entities—for example, specialized roles originating from a single generic role.

Modelers, including senior developers and different levels of managers, must constantly validate the entire modeling system to ensure a faithful representation of the model’s real-world counterpart.

The validation procedure focuses on the active part of the model: the costing mechanism. We call this part active because it uses real data coming from the field. In preparation for it, modelers build two additional diagrams: the infrastructure hierarchy and the infrastructure-activity diagram. These diagrams identify and decompose the infrastructures used in each activity, so that modelers can account for the costs of each activity.

**Validating the model**

The validation cycle for AB-OOBMC relies on a specific adaptation of the Activity-Based Costing technique (Gary Cokins, *Activity-Based Cost Management: Making It Work: A Manager’s Guide to Implementing and Sustaining an Effective ABC System*, Irwin Professional Publisher, Chicago, 1996). A BC manages to account for variable costs by dividing a process into elementary activities, then tracking the time spent in each activity by each person. However, A BC does not specify any mechanism for eliciting activity decomposition. This is why the first modeling steps of AB-OOBMC provide a complete activity diagram. These diagrams become the basic activity diagram for ABC, producing a personalized activity log for each person participating in a process. The activity log, compiled daily, shows the amount of time spent in every activity. Two special cases are possible: Someone can

- consistently spend no time on a particular activity or
- spend a certain amount of time in an activity that does not exist.

In the first case, the activity must be verified. It may be that the particular activity is not performed at all times, but is still part of a particular life cycle development process. Sometimes, however, the model is wrong, and a particular activity does not adequately depict the situation. In this case, modelers may need to conduct further interviews to correct the model.

In the second case, the model is probably wrong and must be updated. A cross-check and subsequent set of interviews will determine the amount of change needed.

This apparently simple check tends to keep the model up-to-date with the actual work performed, eliminating possible management delusions that might obscure the real job being carried on daily.

**AN AB-OOBMC INDUSTRY CASE STUDY**

We developed AB-OOBMC in an academic environment. Although an engineering faculty can coordinate closely with industry, nothing can replace a workplace validation. For this reason, we arranged to test our method at an Italian software company, Engineering Ingegneria Informatica, a subsidiary of Engineering S.p.A.

**Background**

A distributed company, Engineering Ingegneria Informatica divides every project into smaller tasks, then assigns them to units that, in many cases, are located far apart. EII’s core mission is software development for business and public administration systems, a job at which they excel. The company dominates the market in public administration software.

EII management anticipated that applying our method would reveal the complete picture of a distributed software development process, allowing them to optimize the process. Further, the A BC part of our model showed promise for comparing the figures generated by the accounting department with the cost figures obtained directly from employees. This comparison let EII obtain two different estimates of its standard personnel costs.

Top management at EII firmly believes that process improvement must be continuous, and thus willingly made a significant investment in uncovering the current process so that they could improve it. For this reason, they chose to test our method by modeling a project that involved a municipality client and three distinct units as the providers.
Figure 4. An activity-roles-people snapshot for Engineering Ingegneria Informatica’s package division. The circles represent activities, while FORN, OES, OES+, PRO, CSE, and OPE are abbreviations for the roles performed.
Progress

We introduced AB-OOBM C at the management level with a series of seminars that first explained the pilot project’s objectives, then the method itself. After managers felt comfortable with the theory, an adaptation phase followed, in which we tailored AB-OOBM C to the environment at EII. Through a training course, we then taught everyone involved in the project this refined method. At the beginning of the course, management explained the whole project, its main objectives, and the AB-OOBM C approach. A few individuals then underwent separate training as modelers, which included more details about AB-OOBM C.

We experienced problems with some employees, who considered keeping a daily log a boring and useless activity. To alleviate this issue, we designed personalized time sheets that had been preprinted with as much information as possible.

The modelers raised another issue. Without some sort of automated system, they observed, tracking of the model would be virtually impossible. Fortunately, EII had already distributed an easily configurable business process reengineering tool. The modelers designed a tailored version of this BPR tool to support AB-OOBM C, then adopted it as the project’s standard graphic support tool.

Still, the resulting method had to bend to accommodate reality. This adaptation meant slightly changing the modeling procedure’s symbols and mechanics to better fit employees’ requirements. Icons were introduced for standard activities. Secretaries were asked to fill those forms dealing with “institutional” activities, such as general meetings, union councils, and so on. Three months after the first meeting, the snapshot shown in Figure 4 became available for review. (To protect the employees’ privacy, we have changed their names.)

Once the model became available, project members could print the ABC sheets for model verification. Table 1 shows a snapshot of the macroactivities that resulted from this analysis. These results show that the project’s managers had suggested the right people for building the use cases. We needed to change only a few activities. In particular, one missing activity—telephone calls—had great relevance. Italian companies regard phone use as unproductive. In the case of a geographically distributed company, however, phone conversations are not only necessary, they are also much more time- and cost-efficient than travel.

Nevertheless, employees hesitated to account for their phone time, fearing they would be regarded as unproductive. This behavior resulted in the first model underestimating the communication overhead that occurred among units.

After six more months, we considered the model stable enough to analyze trends based on them. These analyses revealed some significant findings. For example, an activity performed by one unit seemed inefficient compared to how quickly another unit performed the corresponding activity. The activity dealt with managing customers’ requests for changes. To understand what customers wanted changed and why, the two groups used two different techniques, one of which appeared much more effective than the other. A more thorough analysis revealed a communication problem, shown in Figure 5, that introduced redundancy in two units, but left the third unaffected.

In another case, two seemingly unrelated activities—the customization activity and ad hoc development activity in Figure 6—presented almost identical opposite trends. Further investigation revealed that the project leader actively guided both, which meant that one activity could receive resources to progress only at the expense of the other because of suboptimal resource allocation. A better allocation of resources would have eliminated this problem, since it was not due to the nature of the tasks.

The modified AB-OOBM C provided sufficient insight for optimization analyses, trend plots, and project control. Although the modeling effort stopped when the project ended, managers felt satisfied with the results. The optimizations introduced through AB-OOBM C have become part of EII’s corporate culture, and helped the company realize its goal of continuous improvement.

A B-OOBM C has passed its first road test, but we modified and simplified it to do so. Many of the lessons learned in this project dealt with adequate manage-
ment commitment to the effort and allocation of resources for A B - O B M C training and implementation. Although concern about divulging proprietary information prohibits us from revealing specifics, we can acknowledge that the project’s modeling activities consumed resources and that the return on this investment was not immediate. Rather, the benefits showed over time, in an improved business culture that builds easier processes, assigns lighter tasks, and makes more precise estimates.

Our approach’s key advantages include the precise and public knowledge of the process activities, the clarification of roles with that of responsibilities, and the buildup of a global-process vision that fosters personal optimizations. The E II modeling project provided a meeting point for academia and industry on grounds that, we hope, will extend to encompass ever wider horizons.

**Giancarlo Succi** is a professor in the Department of Electrical and Computer Engineering at the University of Alberta. Contact him at Giancarlo.Succi@ee.ualberta.ca.

**Luigi Benedicenti** is an associate professor in the Faculty of Engineering at the University of Regina. Contact him at Luigi.Benedicenti@uregina.ca.

**Stefano De Panfilis** is a manager at Engineering Informatica, S.p.A., Rome, Italy. Contact him at depa@mail.eng.it.

**Tullio Vernazza** is an associate professor in the Dipartimento di Informatica, Sistemistica e Telematica at the Università di Genova. Contact him at tullio@dist.unige.it.

---

**Figure 5.** Communications problem revealed through trend analysis. The first unit (a) pursued the assigned activity with no redundancy, (b) the second unit allocated two people, while (c) the third unit allocated three.

**Figure 6.** This activity trend analysis revealed that one manager divided his attention between two concurrent but unrelated activities: the ad hoc activity, shown in blue, and the customizations activity, shown in orange.