ARCHITECTURE &GOVERNANCE magazine

VOLUME 5 ISSUE 11

BUSINESS CAPABILITY MODELING: Building the Hierarchy

By Len Greski

In *Business Capability Modeling: Theory and Practice*, we introduced a set of techniques for modeling business capabilities: unique combinations of people, business processes, and physical assets that deliver value to customers or shareholders. The business capability model is a useful abstraction that connects business strategy to the future business architecture. The capability model also provides a way to organize and prioritize investments in physical assets: buildings, equipment, and information systems. We outlined four major steps to build a capability model, including:

1. Develop the capability hierarchy.

MORE ON PAGE 2

ARCHITECTURE
&GOVERNANCE
magazine

PUBLISHER
Matt Price

EDITOR-IN-CHIEF
George S. Paras

MANAGING EDITOR
Holt Hackney

ART DIRECTION & DESIGN
BNewton Associates

Architecture & Governance
8601 RR 2222
Building 3, Suite 300
Austin, TX 78730
512-536-6270
www.ArchitectureandGovernance.com

Architecture & Governance Magazine is a publication of Troux Technologies. Entire contents © 2009 Troux Technologies. All rights reserved. Reproduction of this publication in any form without prior written permission is forbidden. The information contained herein has been obtained from sources believed to be reliable. Troux Technologies disclaims all warranties as to the accuracy, completeness, or adequacy of such information. The opinions expressed herein are those of the authors, and are subject to change without notice.

2. Identify key relationships.
3. Develop the demand (utilization) model.
4. Develop the financial model.

In this article, we will focus on the first two steps in the process of developing the capability model.

DEVELOP THE HIERARCHY

Before starting a modeling activity, your team must decide on the scope of work to be undertaken. The largest scope is the enterprise. However, capability modeling is relevant wherever an organization unit has profit and loss responsibility, or where a department manages capabilities to which both benefits and costs can be directly attributed. Geographic business unit, division, product/service line, and profit center are all levels of scope where the technique can add value. Work with business partners to identify a business owner for the modeling exercise, ideally the leader who has P&L responsibility for the scope being modeled, or a strategic planner for the P&L area. Once a scope is identified, proceed through the following steps to develop the hierarchy.

> Before starting a modeling activity, your team must decide on the scope of work to be undertaken.

1. Gather information about the capabilities.
2. Draw the hierarchy.
3. Validate the hierarchy.

Gather Information

Work with your business partner to review existing sources of information, including business planning deliverables, marketing materials, public documents about your organization, and models such as business processes, financial plans, and expense forecasts. When reviewing the existing information, identify things that are presented as offers to external customers or constituents (either products or services) or are listed as revenue or cost centers within the financial models. Draft these items into an initial list of candidate business capabilities. Identify the stakeholders within the organization who are responsible for managing these items. Review the candidate capability and stakeholder lists with the business owner as an initial validation step.

Next, interview each stakeholder to learn how he or she talks about the capabilities. What names are used to describe the capabilities? Are there items in the candidate capability list that the stakeholder does not see as significant capabilities? Is it helpful to break down the capabilities into component parts or parent-child relationships? What are the things that drive demand or utilization of these capabilities? How does the stakeholder measure the value

MORE ON PAGE 3

of these capabilities? Which capabilities are customer-facing? Are there any operational capabilities (those that don't directly provide value to external customers) that are necessary to operate this segment of business?

Draw the Hierarchy

Once the interviews are complete, the next step is to organize the information into a hierarchical diagram. Use the root node of the diagram to describe the scope of the hierarchy, in the language of the business owner. The next level of detail should represent the major capabilities within the scope managed by the business owner. If the hierarchy includes both customer-facing and operational capabilities, you may use these terms as the second level in the hierarchy, and organize the major capabilities as the third level of detail. Any one of a number of simple tools can be used to support creation of the hierarchy, including an outline within a word processing document, Post-It® notes, or a diagramming tool that supports organization charts.

Validate the Hierarchy

First, review each level of detail within the hierarchy and confirm that the capabilities within a given level are stated at a consistent level of abstraction. Next, level the diagram by decomposing the initial capabilities into two or three additional levels of detail. Decide on a manageable level of detail (e.g., two to four levels) where each leaf node capability generates discernable benefits and costs, and break down each major capability to the same level of detail. Third, confirm that all parent nodes in the diagram contain at least two children. If a parent node contains a single child, either a second child capability is missing and needs identification, or the parent is sufficiently detailed on its own.

At this point the hierarchy is ready to be reviewed by business stakeholders. Meet with your stakeholders and ask the following questions: Do the capability names use language that is consistent with the way stakeholders speak about the business? Does the organization of the capabilities make sense from a business perspective? Are the parent/child relationships valid? That is, would the business stakeholders decompose the parent capabilities in the same manner as done in the draft hierarchy? Modify the hierarchy to accommodate stakeholder feedback and then conduct a second round of reviews to ensure stakeholder feedback has been correctly incorporated into the model.

IDENTIFY KEY RELATIONSHIPS

A business capability consists of three components: business processes, people, and physical assets. In this step, the modeling team develops association matrices illustrating the relationships between capabilities in the hierarchy and their components. It is also important to map the capabilities to the organization's strategy elements. Often the inputs for the component elements can be harvested from existing documentation, reducing the amount of work for this step.

The relationships can be documented in a planning tool that supports association matrices or with spreadsheets. If the team decides to use spreadsheets, the first step is to create a spreadsheet that lists the hierarchy of capabilities and use this as a template. Insert additional sheets in the spreadsheet for each type of relationship to be mapped to capabilities. For each element to be mapped in this manner, take the following steps:

1. Create a list of the planning elements (processes, people, physical assets, etc.).
2. Add the planning elements to the spreadsheet as a set of column headings.
3. Document the relationships with capabilities by marking the appropriate spreadsheet cells.
4. Identify inconsistencies in the model.
5. Review the association matrices with stakeholders.

Model inconsistencies occur when either the row or column element has no relationship with the other element in the matrix, such as capabilities with no business

MORE ON PAGE 4

processes, business processes with no capabilities, organizational units with no capabilities, etc. List the inconsistencies and work with stakeholders to resolve them.

PRACTICAL CONSIDERATIONS

As a team builds a capability model, it makes a number of decisions to guide the modeling effort. The first decision to be made involves scope of the model. Two approaches can be taken here: 1) start with the largest scope possible, or 2) scope the model to the responsibilities of the primary business stakeholder. Starting with an enterprise scope allows the team to think holistically about the capabilities of the organization and to identify redundant or conflicting capabilities across organization units. On the other hand, scoping to the responsibilities of a business stakeholder aligns the model with a business leader who has P&L responsibility for the capabilities being modeled.

The next major decision faced by the modeling team is the level of detail. Breadth is more important than depth, so try to limit the hierarchy to four or five levels of detail. Manage the modeling to a consistent level of detail, so the leaf-node elements are at the same level of abstraction.

Finally, there are a number of creative approaches to developing stakeholder support for the technique. One approach is to create a straw capability model and socialize it with stakeholders as part of the organization's strategic planning process. This approach allows the team to solicit feedback about its understanding of the capabilities without undertaking a highly visible modeling effort. Another approach is to start the modeling effort when a business leader needs help making investment decisions and scope the model to the capabilities associated with the investment decisions. If the capability model helps a business leader visualize his or her business in a new way, it will quickly gain acceptance. **A&G**

LEONARD GRESKI is the director of eCommerce Architecture at W.W. Grainger, Inc. He can be reached at leonard.m@greski.com.

A&G CALENDAR

Gartner Enterprise Architecture Foundation Seminar 2009
December 10–11, 2009
Las Vegas, NV
www.gartner.com/us/epas4

Forrester's Enterprise Architecture Forum 2010
February 11–12, 2010
San Diego, CA
www.forrester.com/eaforum2010

Forrester's Enterprise Architecture Forum EMEA 2010
March 2–3, 2010
London
www.forrester.com/events/eventdetail?eventID=2439

Gartner Enterprise Architecture Summit
April 14–16, 2010
Las Vegas, NV
www.gartner.com/us/ea

Forrester's IT Forum 2010
May 26–28, 2010
Las Vegas, NV
www.forrester.com/itforum2010

Forrester's IT Forum EMEA 2010
June 9–11, 2010
Lisbon, Portugal
www.forrester.com/events/eventdetail?eventID=2441

Submit your Calendar Events to
editor@ArchitectureandGovernance.com

A&G PODCASTS

Make sure to check our latest podcasts online at www.architectureandgovernance.com including:

- **Craig Martin,** CTO of Enterprise Architects, talks about the evolving roles and responsibilities of the enterprise architect.
- **Dave Sanford,** security strategist and founder of Austin Computer Security, discusses the biggest threats that architects need to have on their radar over the next few years.
- **Philip Allega,** research vice president at Gartner and enterprise architecture analyst, outlines recent trends in EA support for the C-Suite and covers the five key ways the CEO can use EA to support the business.
- **Frank Malta,** executive director of EA and chief architect at Merck, discusses enterprise business capability modeling at Merck and provides advice to EA teams looking to get traction on their EA efforts.

Complementing TOGAF
with the EAM Pattern Catalog

By Sabine Buckl, Alexander M. Ernst, and Christian M. Schweda

Today's enterprises are more than ever forced both to react to environmental changes resulting from a staggering globalized economy and to proactively seek new business opportunities. These aspects demand that an enterprise not only has its business—and IT—functions aligned, but also has incorporated management policies that increase overall flexibility and agility. To enable this enterprise agility, managers must know what their company is good at and which areas retain potential for improvement.

However, this question cannot be answered by thinking in terms of "business-only" or "IT-only." Instead, it has to account for the complex web of interdependencies between business and IT aspects, which together form the enterprise architecture (EA). This architecture constantly evolves over time, as various business and IT projects keep changing the structure of the enterprise and hence tune the interplay of business and IT.

It is necessary to manage this evolution holistically to avoid growing dissonances and to increase the alignment of business and IT. This is why many enterprises are introducing integrated EA management (EAM).

The design of an EAM function for an enterprise is no easy task. Various frameworks exist, as well as EA management tools, which promise to deliver guidance for performing EA management. Still, the presented approaches are either too abstract to provide realization support or far too generic. Hence, they neglect enterprise-specific EA concerns. In this article, we discuss The Open Group Architecture Framework (TOGAF) and its promising but nevertheless highly generic Architecture Development Method (ADM). Complementing the method, we introduce a pattern-based approach to EA management providing guidance for addressing specific EA-related concerns with step-by-step methodologies as well as with corresponding viewpoints and information models.

TWO PROMISING APPROACHES TO START AN EA MANAGEMENT ENDEAVOR

Being confronted with the task to establish an EA management in an enterprise, the responsible architect typically seeks for standards or best practices available in this field. Probably, the most prominent framework for EA management today is TOGAF[1], which was released in its version 9 in February 2009. The core contribution of TOGAF is the ADM, which describes a reference method containing 10 generic phases of EA development (see figure 1). Each phase of the ADM is described by objectives, inputs, steps, and outputs. Thereby, the descriptions stay on an abstract and generic level.

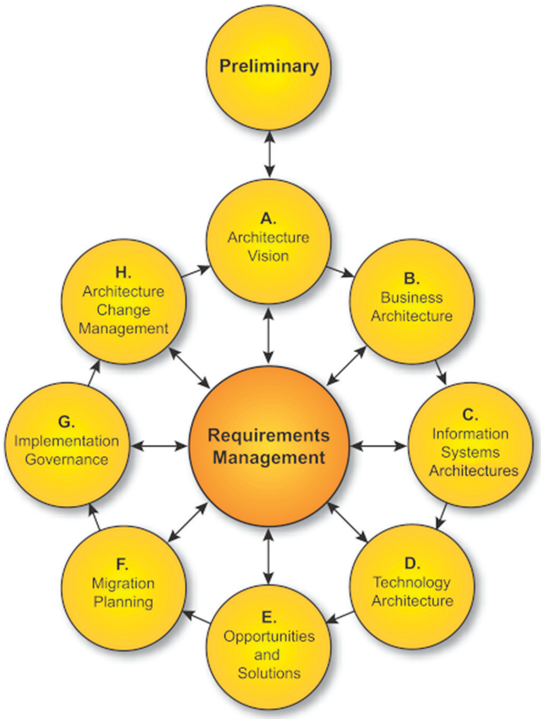

Preliminary
A. Architecture Vision
B. Business Architecture
C. Information Systems Architectures
D. Technology Architecture
E. Opportunities and Solutions
F. Migration Planning
G. Implementation Governance
H. Architecture Change Management
Requirements Management

Figure 1: The ADM cycle of TOGAF

The information objects to be documented, maintained, and visualized are described in the *content framework* of TOGAF. The content framework provides a conceptual information model for describing architectural artifacts, i.e., for storing information to support EA management. This model is designed in a highly general way and, consequently, covers a broad variety of EA-relevant concepts on a fairly abstract level. Nevertheless, TOGAF 9 does not regard the information model to be

MORE ON PAGE 6

[1] See www.togaf.org.

compulsory and explicitly accounts for the combined usage with other conceptual information models.

The Enterprise Architecture Management Pattern Catalog (EAMPC) is another prominent approach, which aims at supporting enterprises in the design of an enterprise-specific EA management function. Patterns have a long history as a means to document proven practice-based, general, and reusable solutions to recurring problems. Their initial exposition dates back to 1977, when C. Alexander proposed the concept in his famous book *A Pattern Language: Towns, Buildings, Construction*. In this book, he introduced the general notion of patterns in the field of architecture, but the concept was quickly adopted by various other disciplines, such as software engineering. The EAMPC builds on the success story of patterns and applies them to the field of EA management. It introduces four different types of patterns:

- **Methodology Patterns (M-Pattern)** define processes and roles to be taken in order to address a given concern.
- **Viewpoint Patterns (V-Pattern)** provide a language used by one or more M-Patterns and propose ways to present data (graphical, tabular, text based, etc.) stored according to one or more Information Model Patterns.
- **Information Model Patterns (I-Pattern)** supply an underlying model and a glossary of the used concepts for the data visualized in one or more V-Patterns.
- **Anti Patterns** document recurring solutions, which have proven not to work in practice in order to prevent typical mistakes in EA management

In addition, EAM patterns explicate potentially conflicting forces influencing the solution for a concern. They further denote known usages in practice as well as implementations of the pattern in EAM tools. Finally, applying a specific solution to a given problem commonly does not come without cost. The EAM patterns account for this fact by giving indications on consequences, which may arise from the usage of the pattern, such as the effort related to data collection. On the level of detail as accommodated by the patterns, an overall process description for performing EA management is not that easy to establish. At this point, TOGAF's ADM provides a valuable complement.

USING THE PATTERN CATALOG TO DETAIL THE TOGAF ADM

Let's look at the single phases of the ADM cycle, the steps conducted in each phase, and how the EAMPC can be used to complement the generic method of TOGAF based on best practice solutions gathered from practitioners and academia.

The TOGAF ADM cycle starts with the *Preliminary Phase*, which prepares and initializes the EA management approach. Typical tasks executed in this phase include the definition and establishment of the EA team, the selection and implementation of supporting tools, as well as the definition of architecture guidelines and principles.

After the preparation and initialization activities are performed, the scope of the EA management endeavor is defined within the *Architecture Vision Phase* (Phase A). A core objective of this phase is to identify the relevant stakeholders and their concerns. Whereas TOGAF details on the management of stakeholders and explicates categories of stakeholders, e.g., *executives*, *line manager*, and *business process experts for the project organization*, it does not give a procedure on how to gather the relevant concerns. Based on the identified stakeholders and their concerns, a high-level architecture vision of the enterprise is derived in this phase. The EAMPC can be used to support the identification of relevant concerns; it explicitly lists typical concerns in the context of EA management. These concerns can be used as input to the stakeholders identified according to TOGAF.

Based on the architecture vision developed in the Phase A, the corresponding business, information systems, and technology architecture is developed in the following phases: *Business Architecture* (Phase B), *Information Systems Architectures* (Phase C), *and Technology Architecture* (Phase D). The fundamental makeup of these three phases is similar: Initially, the baseline architecture is described. Based on this architecture, the target architecture is developed taking the architecture vision into account. This vision was formulated in the preceding phase. Furthermore, a gap analysis is performed to evaluate the differences between the baseline and the target architecture. Whereas, TOGAF describes only a generic EA management process, which needs to be adapted to the specific needs of an enterprise, the EAMPC can be used for this adaptation and to solve the following problems:

- **Identify concepts to be collected:** No information about the exact data to be gathered within the Phases B to D is given. Nevertheless, the importance of gathering only the necessary information to

MORE ON PAGE 7

avoid gratuitous effort is referred to by advice given in TOGAF: *Gather and analyze only that information that allows informed decisions to be made relevant to the scope of this architecture effort*. The EAMPC can be used here to derive the needed information from the concerns identified in the Architecture Vision Phase. In contrast to TOGAF, the EAMPC follows a concern-driven approach and supports the deduction of relevant information model fragments corresponding to the selected concerns.

- **Determine overall modeling process:** To perform this task, best practice solutions as documented in the M-Patterns of the EAMPC can be used.
- **Identify required visualizations:** TOGAF details the importance of choosing the appropriate viewpoints to ensure that the concerns of the stakeholders are covered. Furthermore, the viewpoints need to be selected according to their appropriateness for the stakeholders involved. The EAMPC can be used to ensure the suitability of models and viewpoints as it directly links V-Patterns to concerns.

The *Opportunities and Solutions Phase* (Phase E) is concerned with the derivation of projects and programs, which describe the transformation from the baseline to the target architecture via intermediate transition architectures. The major steps to be performed in this phase are the consolidation of the gap analyses from phases B to D; the identification, refinement, and validation of dependencies between the different architectural layers; and the derivation of transition architectures, which group projects and portfolios. The deduction of transition states can be supported by the EA best practice solutions as documented in the EAMPC. Failure Propagation Specific Proposal Comparison, for example, documents how to automatically generate possible transition architectures in the context of failure propagation.

The aforementioned transition architectures form the input of the *Migration Planning Phase* (Phase F), which is concerned with formulating an implementation and migration plan that schedules and realizes some or all of the transition architectures. The major steps within this phase are the assignment of a business value to each project, the prioritization of projects, and the generation of a road map and migration plan. Within the context of migration planning, the aspect of time-dependence arises, leading to certain demands regarding an information model and the visualizations used in this phase. These demands are addressed by the EAM patterns Functionality Migration Road Map and Business Support Migration of the EAMPC, which can be used to support this phase.

In the *Implementation Governance Phase* (Phase G), the projects selected for realization in Phase F are executed. Major tasks of this phase are identifying deployment resources and skills, monitoring the execution, and conducting reviews, e.g., regarding architecture compliance. The implementation aspects of concerns are addressed in the EAMPC as part of the EAM patterns, which give indications on implementation details. An example of an implementation detail originating from the context of architectural standardization is discussed in Standard Conformity Management.

The *Architecture Change Management Phase* (Phase H) concludes an ADM cycle and prepares the initiation of another cycle. As part of the phase, the changes of the architecture are assessed. Key tasks of this phase are deploying monitoring techniques for the architecture process, developing change requirements to meet performance targets, and managing the governance process. Existing analysis and metrics patterns from the EAMPC, e.g., Verification and Audit, can be used here to assess the achievement of the specified objectives.

Complementing the generic process of TOGAF with patterns from the EAMPC has been proven to facilitate the conduction of EA management projects in practice. In different projects, we have used EAMPC's concern-driven approach to realize the generic steps of TOGAF's ADM. Especially, the possibility to derive the information model from the concerns risen by the stakeholders provides a useful extension to the ADM. Furthermore, the best-practice visualizations of the EAMPC have been proven to be used to augment the deliverables defined within TOGAF.

The EAM Pattern Catalog and case studies can be found at http://eampc-wiki.systemcartography.info. **A&G**

SABINE BUCKL, ALEXANDER M. ERNST, and **CHRISTIAN M. SCHWEDA** are research assistants at the chair for Software Engineering for Business Information Systems (sebis) at the Technische Universität München (TUM–Munich University of Technology).

In smart architectures, business systems become knowledge companions, enabling never-ending, on-the-job training. Real-time evaluation of business rules enables dynamic processes and personalized, just-in-time delivery of completely up-to-date business guidance. A smart architecture is one equipped to address the formidable challenges facing businesses today. Effective engineering of the point of knowledge (POK) is the new litmus test for agility.

The Point of Knowledge

By Ronald G. Ross

The *point of knowledge* (POK) is a real place. POK is where elements of know-how for business operations are developed, applied, assessed, and ultimately retired. In other words, POK is where business rules happen. Knowledge is power, so you can also think about POK as point of *empowerment*.

POK corresponds to point of sale (POS) in the world of commerce. POK and POS are similar in several ways:

- In both, something is exchanged. In POS, it's goods. In POK, it's knowledge—the know-how of day-to-day business operations.
- In the world of commerce, we often say that the consumer and supplier are *parties* in point-of-sale events. Each of us is a consumer in some point-of-sale events, and many of us act as suppliers in others. The same is true for POK. Each of us is a consumer of know-how in some POK events, and many of us act as suppliers in others. Sometimes we switch roles within minutes or even seconds.

A well-engineered experience at the point of sale has obvious benefits both for the consumer—a positive buying experience—and for the business of the supplier—real-time intelligence about sales volume, cash flow, buying trends, inventory depletion, consumer profiles, etc. A well-engineered experience at the POK likewise has obvious benefits. For the consumer, it means a positive *learning* experience. For the business of the supplier, its benefits include real-time intelligence about the "hit" rate of business rules, patterns of evolving consumer (and supplier) behavior, emergence of compliance risks, etc.

The consumer/supplier experience at the POK is crucial to worker productivity and job satisfaction. In no small measure, optimizing this experience is the real challenge in POK engineering. And it must be deliberate. After all, what's exchanged at the POK is know-how—something you can't carry around in your hands. Nonetheless, your company's know-how is very real. Here is what I mean by *know-how*[1].

> **know-how:** accumulated practical skill or expertness . . . especially: technical knowledge, ability, skill, or expertness of this sort

Today, much of your know-how is tacit—lose the people, you lose the know-how. How can you avoid that? Make the know-how *explicit* as business rules. The POK is how you make that happen. Critical success factors in engineering an effective POK include:

- All communication must be strictly in the language of the business, not IT.
- All interaction must be gauged to the knowledge level (and authorization) of each individual party.

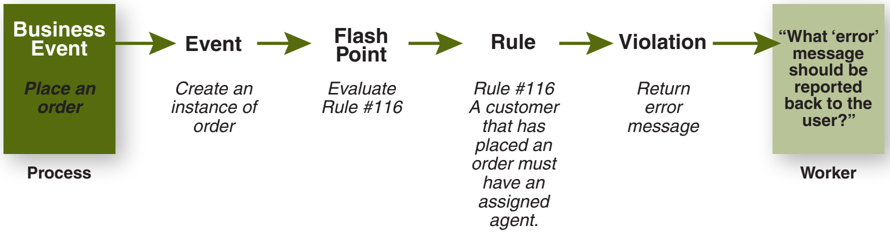
Excerpted from Chapter 10, *Business Rule Concepts: Getting to the Point of Knowledge* (Third Edition), by Ronald G. Ross, August 2009, ISBN 0-941049-07-8, www.brsolutions.com/publications.php. Copyright 2009. Business Rule Solutions, LLC. All rights reserved. Reprinted with permission. For more about the author, see www.RonRoss.info.

[1] From *Merriam-Webster Unabridged Dictionary.*

MORE ON PAGE 9

- Less-experienced parties playing the consumer role must be enabled to perform as closely as possible at the level of the company's most experienced workers.
- Basic business know-how—business rules—must be presented and applied in a succinct, highly selective fashion.
- Basic business know-how—business rules—must also be presented and applied in a *timely* fashion (i.e., "just-in-time") to accommodate fast-paced refinement and change in business policies and practices.

Figure 1: Point of Knowledge Architecture (POKA)


Business Event
Place an order
Process
Event
Create an instance of order
Flash Point
Evaluate Rule #116
Rule
Rule #116 A customer that has placed an order must have an assigned agent.
Violation
Return error message
"What 'error' message should be reported back to the user?"
Worker

POINT-OF-KNOWLEDGE ARCHITECTURE

Let me use an example to sketch the workings of business rules in smart architecture. Refer to figure 1 to visualize how the system works.

Suppose you have a process or procedure that can be performed to take a customer order.

- An order is received. Some kind of event occurs in the system. It doesn't really matter too much what kind of event this is; let's just say the system becomes aware of the new order.
- The event is a *flash point*[2]—an event where one or more particular business rules need to be evaluated. One is: *A customer that has placed an order must have an assigned agent.*
- We want real-time compliance with business policy, so this business rule is evaluated immediately for the order. Again, it doesn't much matter what component in the system does this evaluation; let's just say some component, service, or platform knows how.
- Suppose the customer placing the order does *not* have an assigned agent. The system should detect a fault, a violation of the business rule. In other words, the system should become aware that the business rule is not satisfied by this new state of affairs.
- The system should respond immediately to the fault. In lieu of any smarter response, at the very least it should respond with an appropriate message to someone, perhaps to the order-taker (assuming that worker is authorized and capable).

What exactly should the error message say? Obviously, the message can include all sorts of "help." But the most important thing it should say is what kind of fault has occurred from the business perspective. In other words, it should start off simply by literally saying, "*A customer that has placed an order must have an assigned agent*." We say the error message (or preferably guidance message) *is* the business rule statement.

That's a system putting on a smart face, a knowledge-friendly face, at the very point of knowledge. But it's a two-way street. By flashing business rules in real-time, you have an environment perfectly suited to rapidly identifying opportunities to evolve and improve business practices. The know-how gets meaningful mindshare. That's a ticket to continuous improvement and true agility.

SMARTER AND SMARTER RESPONSES

Is it enough for the system simply to return a business rule statement as the guidance message and stop there? Can't it do more? *Of course*.

MORE ON PAGE 10

[2] For discussion, refer to *Business Rule Concepts: Getting to the Point of Knowledge* (Third Edition), by Ronald G. Ross, August 2009, chapter 8.

For the order-taking scenario, a friendly system would immediately offer the user a means to correct the fault (again assuming the user is authorized and capable). Specifically, the system should offer the user another procedure, pulled up instantaneously, to assign an appropriate agent. If successful, the user could then move on with processing the order.

This smart approach knits procedures together just-in-time based on the flash points of business rules. It dynamically supports highly variable patterns of work, always giving pinpoint responses to business events (not system events). In short, it's exactly the right approach for process models wherever applying know-how is key—which these days, is just about everywhere!

The *Business Rules Manifesto* says this: "Rules define the boundary between acceptable and unacceptable business activity." If you want dynamic processes, you must know exactly where that boundary lies and how to respond to transgressions (at flash points) in real time.

Is that as smart as processes can get? Not yet. Over time, the business rules for assigning appropriate agents might become well enough understood to be captured and made available to the system. Then when a fault occurs, the system can evaluate the business rules to assign an agent automatically. At that point, all this knowledge work gets tucked very neatly under the covers. Even if the business rules you can capture are sufficient for only routine assignments, you're still way ahead in the game.

Smart architecture based on business rules is unsurpassed for *incremental* design, where improvement:

- Focuses on real business know-how, not just better GUIs or dialogues.
- Continues vigorously after deployment, not just during development.
- Occurs at a natural business pace, not constrained to software release cycles.

The *Manifesto* puts it this way: "An effective system can be based on a small number of rules. Additional, more discriminating rules can be subsequently added, so that over time the system becomes smarter." That's exactly what you need for knowledge retention, as well as to move pragmatically toward the knowledge economy. Business rules give you *true* agility.

SMARTER AND SMARTER WORKERS

Now let's talk about smart architecture from a consumer point of view. What people-challenges face your business today? What role should business systems play?

- **Time shock.** As the rate of change accelerates, workers are constantly thrust into new roles and responsibilities. They must be guided through unfamiliar procedures or business know-how as thoroughly and as efficiently as possible. The business pays a price, either directly or indirectly, if getting workers up to speed is too slow (or too painful). Time shock is like culture shock—very disorienting if you're not prepared for rapid immersion.
- **Training.** The flip side of time shock is training—*how* to get workers up to speed. Training is expensive and time consuming. Yet as the rate of change accelerates, more and more (re)training is required. Where do you turn for solutions?

The foremost cause of time shock for business workers is rapid change in the business rules. At any given time, workers might be found at virtually any stage of time shock. Sometimes, you might find them completely up-to-speed, other times completely lost. Most of the time, they are probably somewhere in between. That poses a big challenge with respect to training.

The only approach to training that will truly scale is on-the-job self-training. That requires smart architecture, where pinpoint know-how can be put right in front of workers in real time as the need arises—that is, right at the point of knowledge. What that means, in effect, is that the relevant portion of the company's know-how—its rulebook—is "read" to the worker online, right as the worker bumps up against the business rules.

So a key idea in the business rules paradigm is that operational business systems must become *knowledge companions* for workers in the knowledge economy. After all, isn't making *people* smarter the whole point of knowledge? **A&G**

RONALD G. ROSS is cofounder and principal of Business Rule Solutions, LLC, and executive editor of www.BRCommunity.com.

SPONSORED BY A&G

join the revolution

Cut to the chase and deliver business value faster

FREE ONLINE EDUCATION SERIES

ON-DEMAND **CIO Strategy: Stop the strategy guesswork**
Plug critical management information gaps right away

ON-DEMAND **Business Capability: Stop unstructured business conversations**
Get capability maps and strategic roadmaps in place

DEC 3 **Enterprise Architecture: Stop random EA Modeling**
Begin structured EA Management

DEC 10 **Standards Governance: Stop telling people what not to use**
Empower them to use the right stuff

DEC 17 **Application Portfolios: Stop modeling applications**
Start cutting their costs

SIGN UP NOW AT
www.troux.com/jointherevolution

troux technologies