



Modeling and Code Generation

A Resource For
Extreme Rapid Application Development
(ERAD)

Rationale for BAP

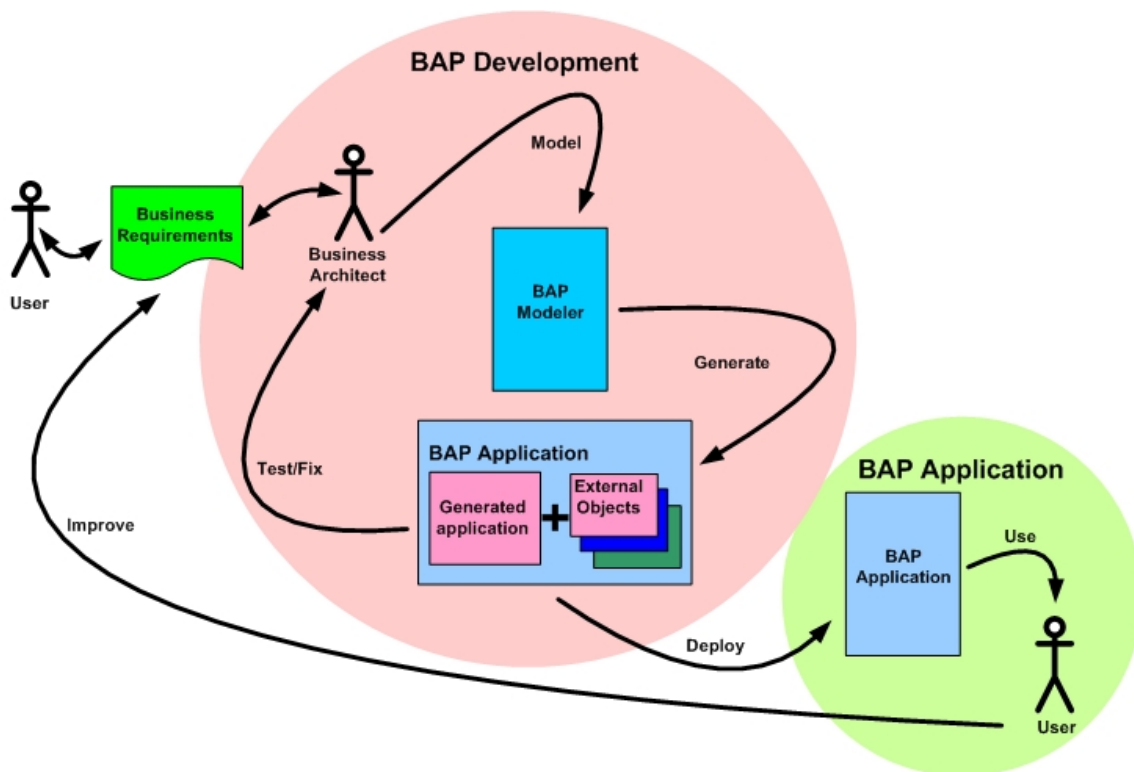
Software systems entail a great deal of complexity. By its very nature, some of this complexity can be qualified as natural. But most of the complexity of any software system is purely capricious. The complexity of a software system is scattered throughout its lifecycle indiscriminately. Most software system development and maintenance projects are risky, lengthily, costly, and more than often, uncontrollable endeavours purely due to their unpredictable complexity. Over the years, the steady growth in the level of end-user expectations, and our dependence based on information systems have led to a far more complicated set of requirements for software systems. The practice of software engineering is a remedy to offset this complexity factor. This practice has paved the way to the birth of new tools, technologies, and methodologies for managing the relentless complexity of software systems.

The primary mandate of the R&D team at Enterprise Applications Platform Inc. is to drastically reduce the complexity of software system lifecycle in general, therefore effectively reduce development risks, timeframe, costs, and provide a framework for

controlling the process. To this end, EAP/R&D has developed a series of concepts, models, and a new development philosophy along with the means to translate the concepts and models to a software system. The cornerstone of model and the philosophy behind it dictates an indiscriminate separation of technology from business and a buffer to absorb the changes in either. These concepts, models, and the philosophy combined resulted in the creation of BAP (Business Application Platform) to be introduced to the software development industry as a new tool and technology for design and development of application programs.

Drastic Reduction in Application Complexity

Software systems are inherently complex. This complexity is due to the nature of the two general categories of requirements; namely direct and indirect requirements. Often, application developers are faced with the enormous task of dealing with a number of requirements that are the indirect results of direct requirements. Furthermore, any software system has a number of resource management and operational requirements. As a result of such requirements, most of the focus of the application developer is subverted to the side issues, and *only a fraction of the application*



developer's resources truly focus on the true business problem at hand.

BAP allows the application developer to relieve oneself from the inherent complexities of the application that have no direct relevance to the actual business at hand. Effectively, the application developer focuses on the business knowledge and how best to represent and model that knowledge in abstract terms. Moreover, BAP furnishes the application developer with a number of pre-programmed business patterns that further simplify and streamline the application development process.

BAP provides a capture tool for the business knowledge. This tool organizes the business knowledge for the application in a generic model, and effectively translates the business knowledge to detailed business specifications. Using a specialized and smart code generator, BAP translates this specification to the actual application, effectively removing many steps in the development process, and hence drastically reducing the complexity and risk involved in the undertaking.

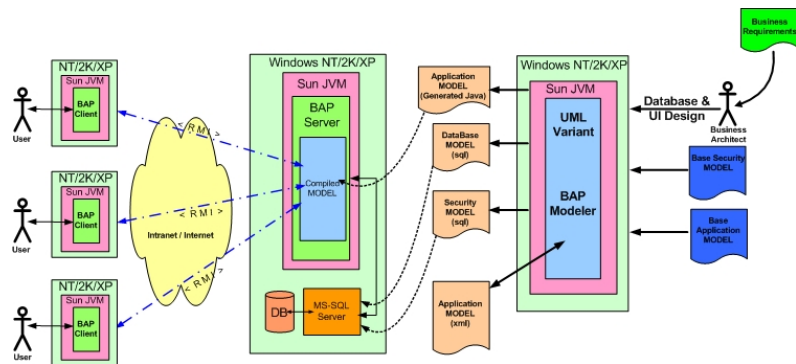
Extreme Rapid Application Development (ERAD)

BAP's business capture tool, Morphware Developer, serves as a RAD IDE. This RAD IDE manages the capture and presentation of business knowledge for the application. It provides a highly generic model for defining business applications. This generic model permits the application developer to concern oneself with the *actual business logic and knowledge rather than the mechanics of implementation of such knowledge*. This concept is stretched to such limits that the developer does not even worry about the screen layout. Effectively, Morphware Developer is truly an extreme RAD tool.

Multiple Design Scenarios

Due to the ERAD nature of the development tool, the application developer can actually model many scenarios of the business and workflow and actually see and demonstrate the real application (not a prototype). This simple attribute of BAP allows the application development company to truly tailor the application to the client's needs at a fraction of time with minimum involvement of expensive resources.

Reliability



The framework on which the actual application is going to run has been tested and verified thoroughly prior to the release of the tool itself. The generator generates the business logic based on the business knowledge entered by the application developer; hence no manual/human intervention is required to “write” the application. These simple statements of fact translate to a highly reliable application program. Effectively, the application developer is not concerned with the traditional debugging and verification process; all the application developer must verify is that one has understood and analyzed client requirements properly. Let the BAP to do the rest.

Loosely Coupled vs. Tightly Coupled

As most software engineers would argue, loosely coupled and component-based systems are much easier to maintain, re-use, and expand, and are generally more reliable than tightly coupled ones. In fact, the only time that a software engineer would consider tight coupling is when performance is of a high degree of importance, and even in that case, the tight coupling is localized to the area where it is required; not throughout the application.

At the same time, application programs that have highly integrated components provide many effortless integration features. In fact, most end users prefer a tight coupling between their application components to ensure that all the parts work with each other seamlessly.

These two totally different approaches require two entirely different development models; namely horizontal development (a.k.a. layered development) and vertical development (a.k.a. feature development). BAP provides for the seamless combination of these two models of coupling in a very elegant way. It simply injects the business logic at generation time into the layers of abstract system logic.

Accurate and Detailed Application Specification Generation



In order for the capture tool to effectively generate an entire application, it must capture and organize the business specifications of the application and map them to the pre-programmed business patterns. In other words, the tool “knows” what the application will do; and that is why it can generate it. At the click of a button, the capture tool generates the entire business (and system) specifications of the application in one of many standard models for documenting specifications.

Project Lifecycle Management

The development tool offers some basic project lifecycle management tools. Workflow and task management are building blocks of the system. Create your “desired” tasks and workflow to automatically take care of the preferred processes and routines in your operations.

Low “Total Cost of Ownership”

Due to the nature of BAP and the low overhead of application knowledge maintenance and change,

total *conservation of investment* in business software development. BAP serves as a buffer between technological changes and business requirement changes, shielding each one from the adverse effects of changes in the other one. But the benefits of this philosophy extend far beyond the “buffering effect.” Other benefits of this approach include (but are not limited to):

- Ability to generate multiple application models from the same specification; a thin-client/fat-server model, a web-based model, a telephony model, a wireless-based model, a PDA model, etc.
- Ability to add a new technology base and generate business applications in the new technology base seamlessly and effortlessly.
- Ability to add new business requirements to the system (extend the business specification) and seamlessly and effortlessly generate multiple models at the same time.

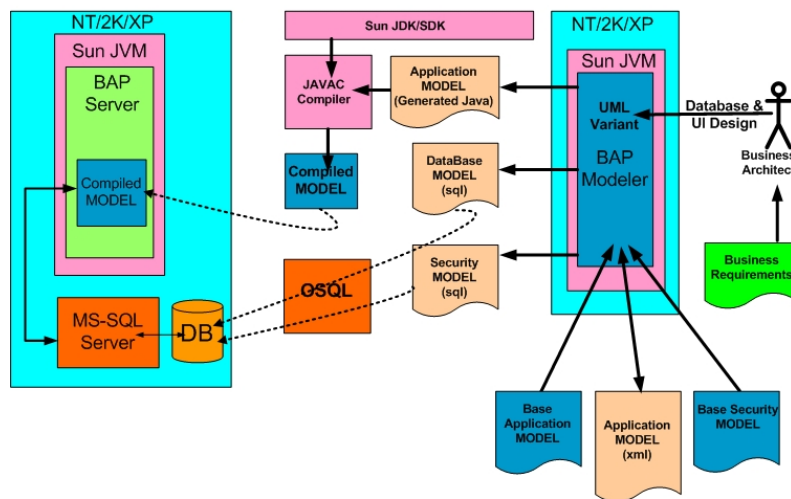
No Need for Traditional Programming

Due to the nature of the development tool, the business knowledge capture process serves as formal programming, eliminating the need for traditional programming; and therefore the need for programmers for the sake of implementing the business logic. The benefits of this attribute are interesting.

One of the factors that reduce the reliability of a business application is the incorrect interpretation of requirements. Requirements are gathered by business analysts, translated into textual representations (in some cases, a formal specification language such as Z is used; but not very practical due to the abstract algebraic nature of such languages), reviewed by

application and system analysts, and finally programmed by programmers who may not have even met the client. In other words, the essence of the requirements are diluted as the requirements are passed from person to person and team to team; furthermore, in many cases, the requirements are polluted by the individual interpretation of implementer.

Please note that if a plug-in is required, an actual programmer must program the plug-in. But even in this case, since the programmer is only concerned with the



business applications designed for this platform enjoy a low total cost of ownership.

Total Separation of Technology from Business Logic

The basic philosophy of BAP is total separation of technology from business logic. This basic model allows the business knowledge to be extended independent of the technology. And the technology base can change completely independent of the business knowledge. This separation translates into

domain of the plug-in and its requirements, the quality of the work is much higher than if the programmer was part of a larger team working on a huge business application program and had to write the same logic.

Parallel Distributed Processing

One of the inherent attributes of BAP is its parallel-distributed processing model. Any application program that uses BAP seamlessly inherits the processing capabilities of BAP. In BAP terms, the “network is the computer”, and using this philosophy, BAP is able to manage a high volume of tasks simultaneously.

Horizontal and Vertical Scalability

BAP is highly horizontally and vertically scalable. Therefore, any business application program based on BAP enjoys seamless horizontal and vertical scalability.

Interoperability

BAP has inherent interoperability features that allow other systems communicate with in a B2B/B2C environment.

Open N-Tier Architecture and Platform

BAP’s architecture and underlying platform is n-tiered. BAP enjoys an open platform, and therefore any

business application based on BAP will enjoy the same benefits.

Single Server, Multiple Client Platform Support

BAP’s server cluster can serve a heterogeneous array of client programs. In other words, while someone is using a business application that runs on a workstation, another user can use a PDA version of the same business application to access the same data.

Database Server Neutrality

BAP works with multiple databases. Effectively, BAP can handle a heterogeneous array of database servers to store and retrieve persistent data. Therefore, any business application program based on BAP can do the same seamlessly; without the need to maintain multiple definitions of stored procedures, etc.

Hardware Architecture and Software Platform Neutral

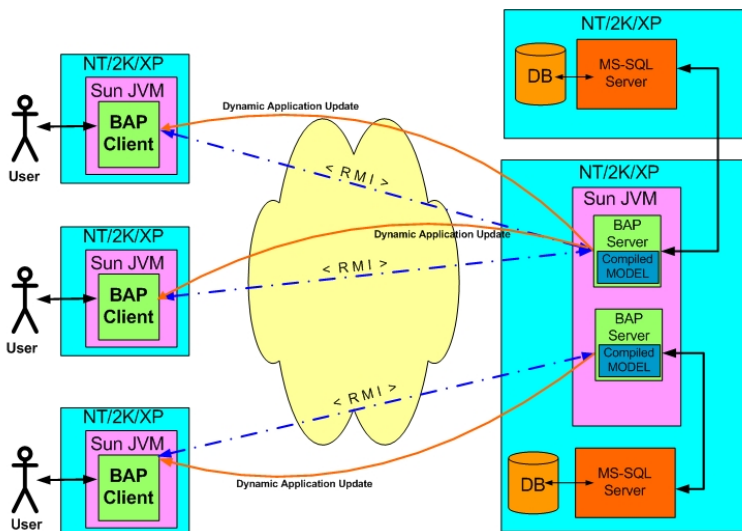
BAP is 100% architecture and platform neutral. Furthermore, the generator modules of BAP can generate code in any language and for any platform (provided that a framework for the target is already programmed).

Documentation

BAP generates up-to-date, accurate, and sufficiently detailed application specification and documentation. The model is the documentation, and the documentation is the program. BAP maintains the system in a model that is easily understood by all developers and literally removes the need for highly technical programmers, lowering the risk of programmer dependency.

Intrinsic Security

BAP provides many levels of security for the business application program from simple application level security to extensive encryption of persistent data and network communication.



The bottom line is - BAP resolves the challenge inherent in the technology vs. business model dichotomy in application development; and in the process provides the ultimate flexibility to respond to changes quickly, reliably and economically.