Pegasystem’s “Insanely Open” Architecture

Pegasystem’s industry-leading Business Process Management (BPM) technology has been recognized by industry analysts and customers for its power to transform the way business and IT collaborate to deliver application solutions. The foundation for this ground-breaking approach to business application development is a robust, “insanely open” architecture that embraces and exploits industry-standards and best practices, while providing a unified, model-driven design environment for business analysts and technologists.

“Insanely open” is more than just a marketing blurb: it is a mantra created and supported by Pega’s core engineering team and proven in over 25 years of delivering enterprise-class solutions.   PegaRULES Process Commander (PRPC), Pega’s BPM and Business Rule Engine (BRE) platform, is built from the ground-up on open standards and industry best-practices to support the service oriented architectures (SOA) that are fast becoming the technology backbone of today’s leading organizations. PRPC’s model-driven development environment enforces a granularity and reuse of business assets unseen in traditional approaches, and allowing for seamless composition and decomposition of processes and business services.   The Java engine at the core of PRPC is proven to support massive degrees of scale and parallelism, provide fine-grained visibility and control for system and security administrators, and allow our customers to extend and customize the platform to fit their needs.

A History of Technology Innovation

PegaRULES Process Commander is the fourth “from scratch” rewrite of Pegasystems technology. Pega was founded in 1983 and initially delivered a PL/1-based solution for DEC VAX machines, allowing organizations like Citibank and Bank of America to design execute and evolve complex exception management process. That solution was rewritten to support VAX and CICS deployments and then later ported to a C++ distributed solution, allowing the growing list of Pegasystems customers to extend their process automation from the back to the front office. In the late 1990’s, Pega engineering began an R&D project to develop the next generation platform, leveraging 15+ years of experience with workflow, case management and business rules, and exploiting the emerging Java and XML technologies that are at the core of today’s enterprise architectures. The result of the development was PegaRULES Process Commander (PRPC), made generally available in 2001 as version 3.2. Pegasystems’ PRPC version 6.1 was released in Q1, 2010.

Built on Open Standards

PRPC is built from the ground-up to leverage open standards. The core of PRPC’s architecture is a Java engine that dynamically assembles caches and executes Java classes based on a repository of XML “business models”: process diagrams, business rule definitions, user screens, etc. PRPC is designed to dynamically adjust to the Java container it is running in, seamlessly exploiting the features of a full JEE application server (WebSphere, Oracle or JBoss), a Java web container (Tomcat) or a simple JVM. PRPC natively supports both XML and Java-based data structures and navigates seamlessly between both, presenting process designers with a single data model dynamically assembled from both XML and Java-based constructs.

PRPC is optimized for Service Oriented Architectures, and can consume and publish services in a variety of transports, including SOAP, HTTP/ReST, JMS, MQ, EJB and Java API's, with both generic services and strongly-typed "business delegate" patterns supported. WS-* standards such as WS-Security, WS-WSDL, WS-Addressing are supported "out of the box." PRPC has supported BPEL since prior to the ratification of BPEL 1.1, and can both import/export BPEL process definitions and leverage BPEL interoperability standards to pass and receive process control to other process engines. PRPC provides a BPMN stencil for diagramming business processes.

PRPC has been designed to fit seamlessly into enterprise application infrastructures. It is built on a JEE architecture to allow for deployment of massively scalable, fault-tolerant systems. Standards-based monitoring (leveraging JMX and mBeans) and world-class alerting mechanisms allow system administrators to see exactly how the system is performing. Role-based and fine-grained access rights and support for LDAP and SSO allow for integration with Enterprise security architectures. PRPC generates user interface elements, leveraging modern, "Web 2.0" technologies such as JSP, AJAX, Flash and XHTML.

Table 1: PRPC's Out of the Box Support for Standards

Security	LDAP, WS-*, JCE-based encryption, SAML, SSL, SSO, Assorted certificate stores
System Monitoring and Management	JMX, MBeans, Apache Logging, Common Event Format (CEF)
Business Monitoring and Reporting	PMML (for import of data analysis); Data export to XML, delimited, fixed and relational formats, SQL-based extension of reporting
User Interface Design and Deployment	JSR-168 and WSRP portlet generation, JSP-based screen generation, JSON-based "mash-ups", AJAX, Flex / Flash
Integration and SOA	SOAP, ReST, MQ, MSMQ, JMS, BPEL, EJB, JavaBeans, JCA, FTP, WSLD import, JNDI introspection
Process Definition and Execution	BPEL, XPDL, BPMN, Visio
Business Rules Design and Execution	Backus-Noor, PMML, JSR-94 for local/remote execution of rules
Data modeling and persistence	XML data models, XSD/DTD import/export, JavaBeans, COBOL Copybooks, XPath and XSL parsing,

Exploiting the Benefits of Model-Driven Development

PRPC provides a modern, model-driven approach to application development. Traditional development approaches rely on non-business friendly "code" as the record of business intent, bind applications to versions at "build time," and depend on developers and architects to design reuse and service-orientation into the application. PRPC's model-driven approach elevates the "business model"—a process diagram, a business rule definition, a screen wireframe, etc.—making the model the record of business intent. This approach provides both business and technology benefits.

- *Increased business and IT collaboration.* Because business and IT work from the same set of models, and because the model will never be manually reinterpreted or exported into a cryptic language, the business intent is never lost and business requirements are always directly aligned with the executing application, allowing business-driven changes and automatic documentation.
- *Dynamic versioning.* PRPC's models are dynamically assembled into Java classes by the runtime engine. Generated code is then cached for reuse across users and across nodes. This approach allows multiple versions of a single model to execute simultaneously in a single environment, even in a single JVM, allowing "grandfathering" of process and rule changes, pilot or "champion/challenger testing" of application changes, and seamless rollback and replay of previous versions.
- *Enforced granularity and reuse.* PRPC's models enforce a level of granularity not present in traditional development. Because process models, rule definitions, screen definitions, etc., are highly atomic, they can be easily reused in the context of different applications. Individual models can also be dynamically exposed as services to be consumed by other applications. Even processes that were never intended to be reused can be dynamically exposed as services, allowing the technology evolve to meet changing business needs.

Leveraging the Power of Java

PRPC is not a proprietary or closed engine. Each model is stored in a defined XML format, transformable to standards (BPEL, XPDL, etc.) whenever possible. At runtime, PRPC generates, compiles, and caches Java code as the execution form of each model. The generated Java is completely visible to authorized users, and PRPC provides extension points that allow developers to inject their own Java code directly into the code generated by PRPC, allowing extensibility to the very of core of the engine.

PRPC's Java-based engine has been able to prove linearity scalability, both horizontally and vertically, to tens of thousands of concurrent users, executing millions of processes and billions of rules concurrently. PRPC can run its engine across numerous leading application servers (WebSphere, Oracle, JBoss) as well as Tomcat web containers. PRPC can execute in a native JVM as well, allowing rules and processes to be embedded inside of other Java applications and batch jobs. The result is that a single process definition can be executed at huge scale across multiple platforms and deployment patterns.

Summary

A single, unified environment for designing and deploying process-based applications enables faster time to market, better collaboration between business and IT, and a rhythm of change that is not possible with solutions that have been cobbled together from separate rules, BPM, UI, and case management products. While BPM systems complement and participate in Service Oriented Architectures, Business Process Management should not be confused with middleware technologies, or even SOA strategies. Rather, BPM represents a new, model-driven approach to delivering and evolving enterprise business applications. PRPC provides a technology platform that is built on open standards, leverages the flexibility and reusability of a pure model-driven development approach, and leverages an open Java-based engine to ensure performance, extensibility and platform interoperability. The result is a modern, open architecture that delivers the agility only possible with a unified technology, and the flexibility and performance that comes from today's leading technologies.