

AgilePoint

Version: 3.2

UPDATED

Ascentn Corporation

1674 N. Shoreline Blvd, Suite 136, Mountain View, CA 94043

Tel: 650-968-6789 Fax: 650-968-6785

Web: www.ascentn.com Email: info@ascentn.com

1 Product Overview

The Ascentn AgilePoint was designed to enable organizations to deliver BPM objectives with the philosophy of *Think Big, Start Small, and Scale Fast*. Ascentn believes this is the most practical way to practice BPM and create managed success in view of the challenges faced with technology, people and culture adaptation.

From the ground up, AgilePoint leverages the entire Microsoft product set and environment (.NET, Visio, InfoPath, SharePoint, ASP.NET, Exchange, Windows Workflow Foundation, Office System OpenXML, SQL Server, BizTalk, and Visual Studio, etc.) to deliver a robust and scalable BPM Engine, which, through reuse of process assets, supports both human workflow and automated processes and enables business managers to leverage the stack of Microsoft products and technologies as simply as diagramming a process model. The product extends Visio into a full lifecycle BPM productivity tool, from design into production, and through iterative improvement. It allows business managers to model and execute business processes straight from Visio, free of rigid code generation that limits run-time adaptability, to deliver a high degree of in-flight process adaptability and hence business agility.

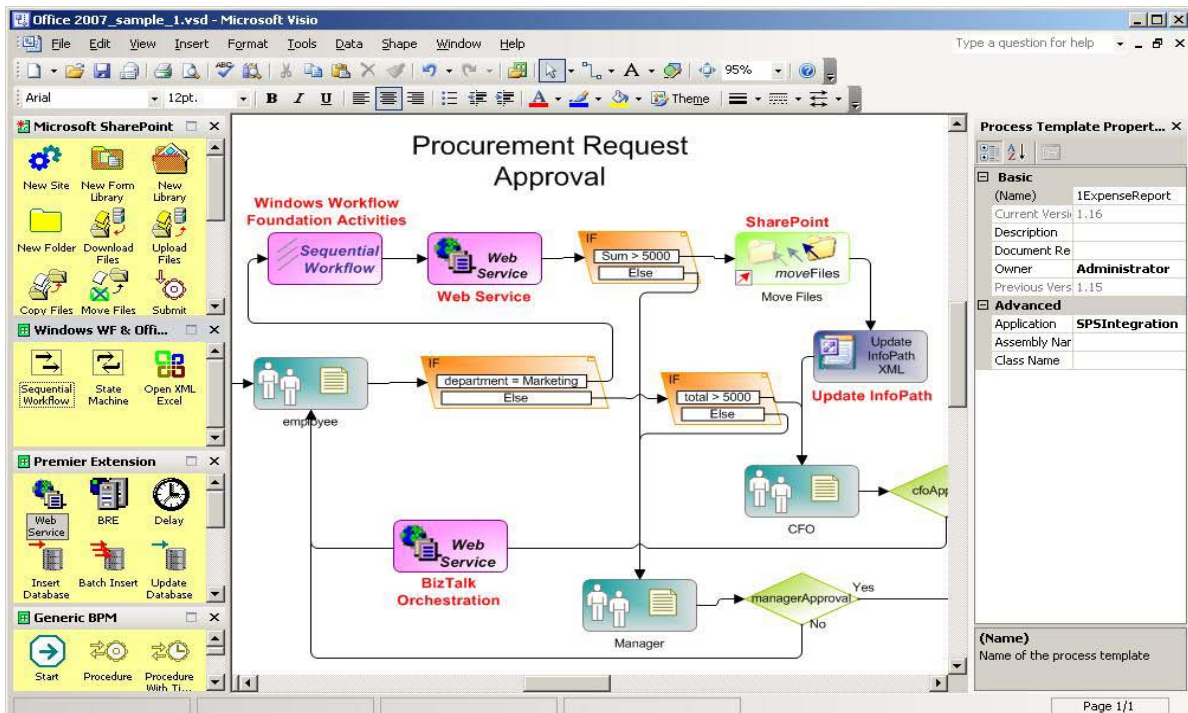


Figure 1. AgilePoint extends Visio into a full lifecycle BPM productivity tool.

AgilePoint is entirely .NET based; aiming to deliver a platform that is 5 to 10 times more cost effective

Ascentn

AgilePoint, Ver. 3.2

than J2EE-based BPM Suites, thus extending affordability to mid-size and small businesses. Although based on .NET, AgilePoint employs a lot of J2EE type functionality with highly scalable discrete components reflecting the past experience Ascentn's key designers have had with J2EE. The result is a highly scalable n-tier process architecture that allows firms to easily abstract process logic from underlying application codes and enables business managers to change and control business processes directly.

Through its tight integration of Visio, AgilePoint can provide end-users with an accessible environment, allowing them to re-use existing process descriptions by adapting them for the AgilePoint environment. Through the integration with Visual Studio, AgilePoint lets IT control the infrastructure, leverage existing Microsoft assets and skill sets, and expose custom and SOA developments in abstracted, easy-to-understand process descriptions. Through its tight integration of Visio, the product provides business managers a familiar and accessible environment and lets them control business processes through re-using existing process descriptions to create or improve directly executable workflows, end-to-end business automations and SOA applications, creating effectively an actionable process layer.

Under the covers, and conforming to WfMC XPDL, AgilePoint is standards-based via its engrained XML support. XML-based process descriptions are fully translatable into emerging standards such as BPEL (once it clarifies). At runtime, the BPM Engine directly consumes XML process models and performs dynamic binding of process fragments via its Web Services functionality, enabling highly adaptable process architectures.

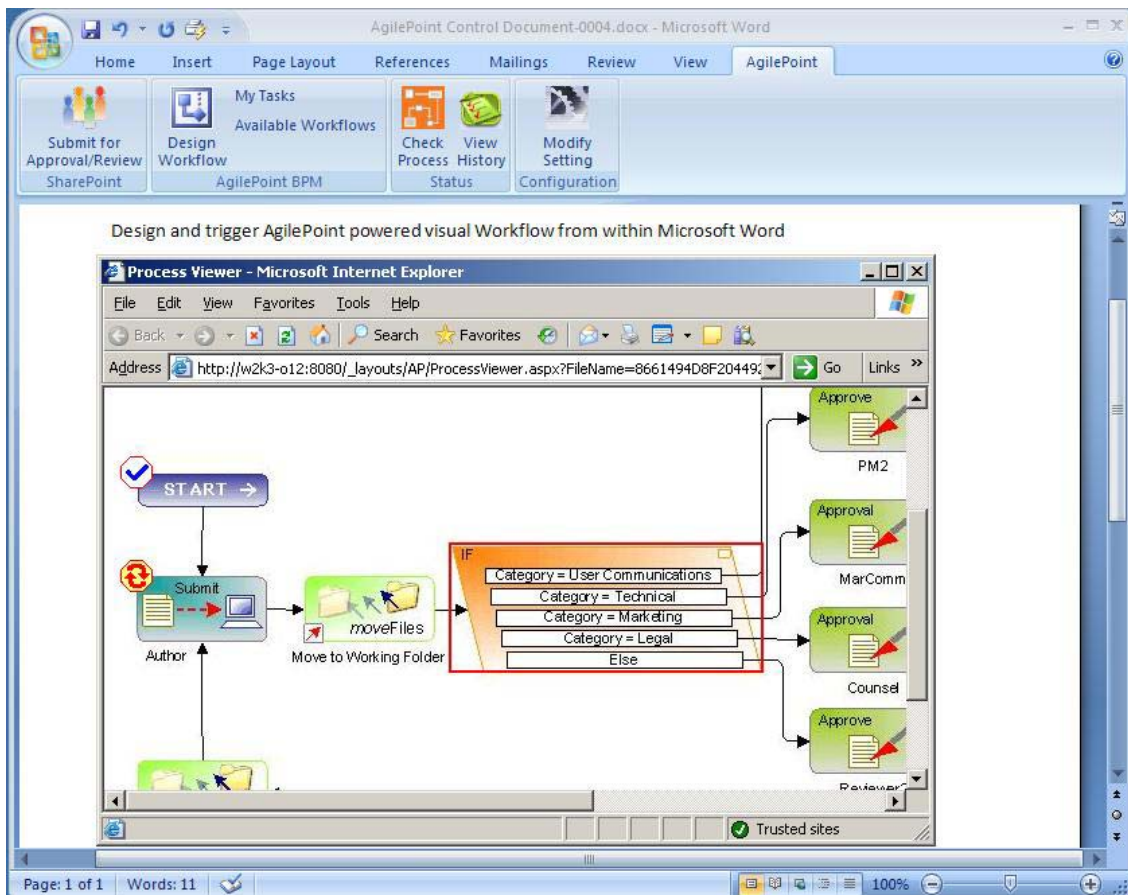


Figure 2. AgilePoint delivers tight integration with Microsoft Office 2007.

Ascentn

AgilePoint, Ver. 3.2

AgilePoint delivers tight integration with Office System 2007 suite. For example, for InfoPath and SharePoint, AgilePoint directly reuses functionality delivered in these environments. For instance, basing a process model in InfoPath allows the model to inherit its underlying data structure and any pre-defined views. With regard to SharePoint, AgilePoint provides integration with document libraries, form libraries, lists, etc. The integration allows business managers to create sophisticated business processes graphically in an unrestricted authoring environment based on Visio. This enhances the native facilities provided by SharePoint, extending the native collaboration and document management capabilities to support cross-functional, end-to-end business processes. AgilePoint effectively elevates the business user empowerment goal of SharePoint to a new level that lets enterprises leverage SharePoint as a highly adaptive 'hub' for their end-to-end business operations.

In addition, AgilePoint also integrates tightly with Microsoft BizTalk for robust back-end information flow orchestration, integration and transaction capabilities to promote reuse and expedite BizTalk-enabled application deployment. BizTalk orchestrations and business rules can be abstracted as easy to use process descriptions and directly included into their process models by business managers. As a result, this allows AgilePoint to drive sophisticated BizTalk orchestrations, and leverage its Business Rules Engine to provide a comprehensive enterprise class BPM platform. Other supported Microsoft integrations include Microsoft Dynamic solutions such as Axapta, CRM, etc. AgilePoint also offers a rapid enablement solution to migrate Lotus Notes workflow to .NET.

2 BPM Engine

The AgilePoint environment is designed to deliver high performance, yet to be lightweight, allowing deployment in virtually any business operation. The advanced event-driven, object-oriented, *pluggable* and *moldable* architecture allows domain specific extension for use by third party ISVs and OEMs. This architecture enables customers to directly leverage AgilePoint engine components, reducing development cost and time. The AgilePoint architecture uses Microsoft .NET and Web Services to link together the presentation layer, logic layer (AgilePoint Server) components, and data layer.

AgilePoint presentation layer provides interfaces for the design, development, execution, management, and monitoring of the process and its instances. AgilePoint incorporates the following presentation components (of course, AgilePoint can also be embedded in external applications):

- *AgilePoint Envision* – an extension of Microsoft Visio, enabling process designers to define and simulate their process templates and deploy for execution through the Microsoft Visio interface
- *AgilePoint Developer* – an extension of Microsoft Visual Studio, allowing application developers to leverage AgilePoint's BPM framework, e.g. AgilePart, and abstract custom developments, Web Services, etc., into reusable process descriptions at the process tier. It also lets firms implement more complex process and application logic to create domain-specific extensions, reusing processes, and other AgilePoint server functionality
- *AgilePoint Enterprise Manager* – a web-based interface for power users or system administrators to manage, monitor, and audit the AgilePoint BPMS and its runtime processes

- *AgilePoint Server Configuration* – a desktop-based configuration tool for the system administrator to configure the various system parameters such as SMTP server, database settings, custom server extensions, etc.

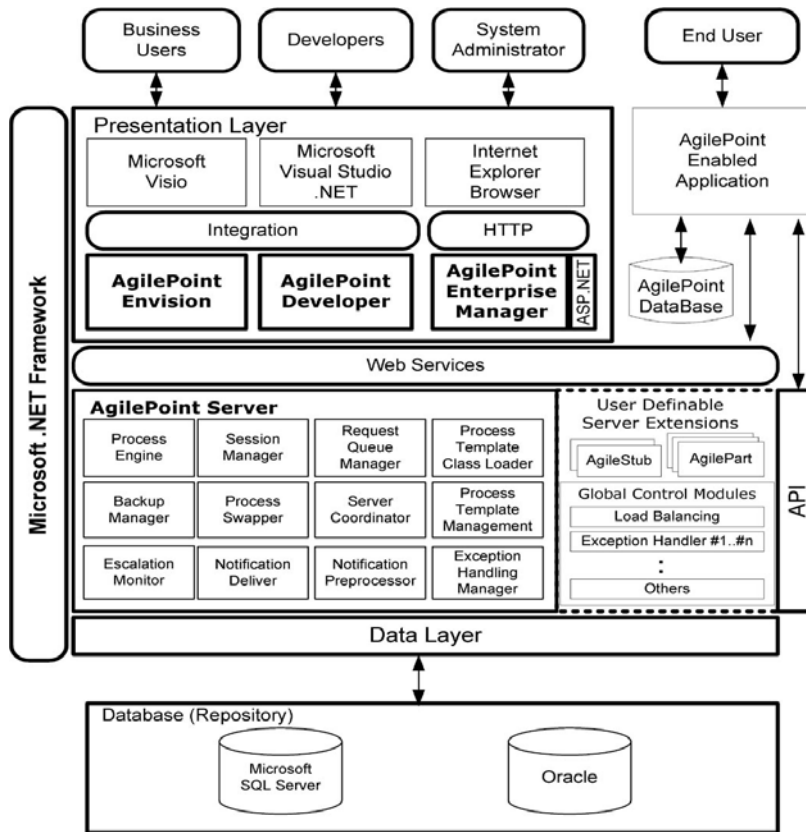


Figure 3. The AgilePoint Architecture is built from the ground up on .NET

The logic layer provides the core functionality of the AgilePoint Server. There are multiple components, each providing its own specific features and functions. These components interact with each other, the presentation, and the data layer. They are invoked either through components in the presentation layer, through the AgilePoint Web Service interface, or through the Server-side API.

The major components are:

- *Process Engine* – provides the core of the AgilePoint Server. It is a WfMC compliance process execution engine using an extended set of XPDL constructs. It consumes XML-based process definitions designed and deployed through Envision. The engine is capable of performing dynamic binding and loading during execution, supporting enhanced flexibility and agility at runtime without incurring engineering cycles.
- *Session Manager* – handles the client application connection session, including security authentication and authorization. This component is also responsible for monitoring the behavior of the activities throughout the process and for keeping track of the audit trail of the process.
- *Request Queue Manager* – manages task requests. It maintains the multi-threaded, asynchronous execution of a series of task requests, which can be prioritized, delegated, reassigned, suspended, resumed, and cancelled. Control over these facilities is either through programmatic rules or pre-defined delegation rules, or they can be triggered manually through the Enterprise

Manager.

- *Process Template Class Loader* – handles the loading of process models and their execution instructions. For example, the instructions could be a Visual Studio.NET project containing custom code to control the runtime behavior of the process instance. The process template definition and instructions are fed to the Process Engine.
- *Process Swapper* – uses an intelligent algorithm to swap out processes from system memory to maximize the use of system resources and offers high degree of efficiency for managing long running processes.
- *Server Coordinator* – manages the server and system configuration, including database configuration; system administrator settings and authentication mechanisms; and email related settings. In a multi-server deployment scenario, this component also manages the clustering and load balancing relationships and configurations among multiple servers.
- *Escalation Monitor* – keeps track of whether an activity or a process is overdue and triggers appropriate action via the Exception Handling component (which also monitors system-level exceptions). For example, AgilePoint can dynamically invoke a *fast-track* subprocess, raise priority, or reassign to a different user for an overdue task.
- *Notification Preprocessor and Deliverer* – handles email notifications. These can be based on different states that an Activity may enter.

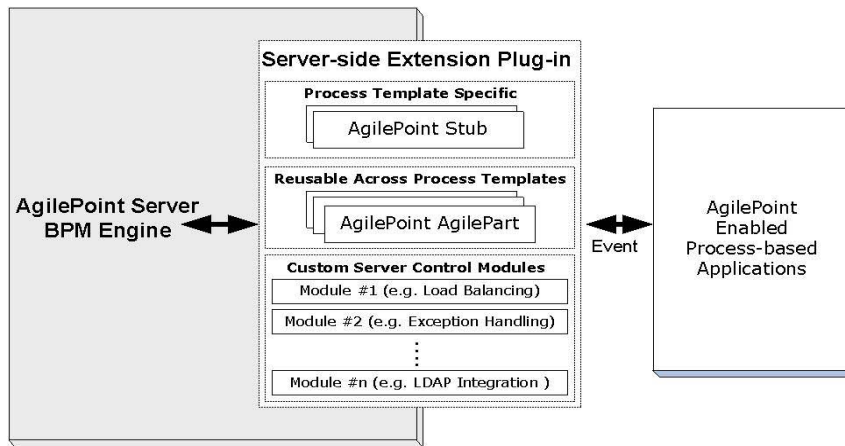


Figure 4. Extending the capabilities of AgilePoint via the server-side plug-in framework

The core functionality of the AgilePoint Server can be extended through the AgilePoint BPM frameworks—AgilePart, AgileWork, AgileStub, and AgileConnector, etc.—to enable abstracted leverage and manageability of applications and infrastructures at the process tier. All are created with AgilePoint Developer via standard AgilePoint enabled project templates and AgilePoint APIs in Visual Studio.

- *AgilePart* provides the integration mechanism to extend AgilePoint at the “activity” level with external applications, such as ERP, CRM, etc., that is configurable and reusable across process templates and can be used by business managers at the process tier to build directly executable process models. AgilePart functionality can be generic or domain-specific. AgilePart can be directly reused by business managers at the process tier, (i.e. within Visio-based AgilePoint Envision, instead of within a developer’s environment). AgileParts can be individually registered with the AgilePoint Envision modeling environment (in Visio) or centrally managed and downloaded as a *group* from the AgilePoint Server. AgilePoint provides out-of-the-box AgileParts for the following applications and technologies: SharePoint, InfoPath, SQL, Web Service, Exchange,

Office System OpenXML, and Windows Workflow Foundation, etc.

- *AgileWork* provides the integration mechanisms to extend AgilePoint at the ‘activity’ level with reusable, configurable meta-data-driven dynamic manual activities that are shareable across process templates.
- *AgileStub* provides the ability to associate external functionality that is intended for a specific process template.
- *AgileConnector* provides users with a way to extend the core functionality of AgilePoint Server or to control various aspects of its runtime behavior. Examples include custom exception handler, custom load balancing, message queues, etc.

Once an AgilePoint process model is deployed to the AgilePoint Server, the Server will instantiate the process model in response to user requests (through the user interface of an AgilePoint enabled application), an event, or programmatically. The AgilePoint Server then executes the process instance based on the definition, while maintaining status information in the database. This control data is used in process management, monitoring, and reporting.

Web Services can be used to call subprocesses (and publish process descriptions). They are also used within AgileParts to integrate external functionality.

2.1 Platforms

AgilePoint server runs on Microsoft Windows 2003 Server and the coming Windows Vista Server. Supported databases are, respectively, SQL Server 2005 and Oracle databases through ADO.NET, and Oracle Data Provider for .NET. This architecture can be easily extended to support other data sources, if needed.

2.2 User Interface



Figure 5. The MyTasks list is delivered in a web browser

AgilePoint application interfaces provide capabilities for users to perform their manual activities and also to manage, monitor, and audit the processes. Users can access and perform their assigned tasks through a variety of interfaces including Outlook Tasks via Exchange Server integration, AgilePoint Enterprise Manager, or embeddable in-application interfaces such as Task Management Webpart for SharePoint, Task Management Control for ASP.NET, etc. Through AgilePoint Enterprise Manager, users have access to a browser-based environment and are presented with a holistic view of all the tasks assigned to them and/or any issues awaiting resolution across all areas. Drilling down into MyTasks takes the user to a list oriented view. A task could be assigned directly to a specific user, or it could be placed in a shared queue that is accessible to all members of a given role or group. Tasks in the queue can be prioritized. Further, AgilePoint is also capable of supporting dynamic, multi-level, parallel process voting.

AgilePoint - Document Control Task List						
Hello, DRAGONFLY-EN\manager						Download AgilePoint Integration For InfoPath
						1/23/2005
Document	Location	Pool	Action	Due Date	View Process	
SalesForecast-3	/sites/AscentnFinancialSite/Financial Control Doc Lib/Working		Approve Cancel Reassign	1/24/2005 10:14:00	View Process	
Exp_Rpt(Manager Approval)	/sites/AscentnFinancialSite/ExpenseReport_InfoPath		Submit Cancel Reassign	1/24/2005 19:54:00	View Process	
Invoice-2	/sites/AscentnFinancialSite/Financial Control Doc Lib	2		1/24/2005 19:57:00	View Process	

Figure 6. The AgilePoint task management user interface as a SharePoint Web Part

AgilePoint also provides task management interfaces that can be easily incorporated into an AgilePoint enabled application. These interfaces include task list, task status, process viewer, etc., and are included automatically when creating an AgilePoint Web or server application in AgilePoint Developer. Of course, the look-and-feel of these interfaces can be further modified to meet custom needs. AgilePoint also provides a task management Web Part for the SharePoint Server that shows only tasks related to SharePoint-based business processes.

2.3 Scalability

AgilePoint has adopted a highly scalable 3-tier architecture, which can scale to n-tier. The decomposable architecture allows AgilePoint’s various components to be deployed into a single server or across multiple servers. AgilePoint supports both local and remote database servers and can share the same database server with other applications.

Through NLB and clustering, AgilePoint provides overall *scale-up* scalability. In addition, AgilePoint also provides application context driven *scale-out* scalability by allowing process fragments and AgileParts to be executed on their own dedicated or clustered server, reducing the load on the main AgilePoint Server environment to ensure predictable process management performance. The *scale-out* scalability is accomplished by deploying a custom *load balancing* AgileConnector to the AgilePoint Server.

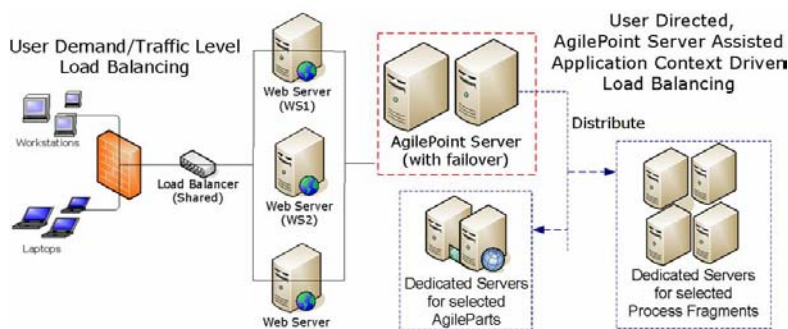


Figure 7. AgilePoint can be distributed and incorporate additional process engines

AgilePoint provides fail-over support by switching to a hot standby back-up server automatically or manually. Automated switching could be accomplished by third party hardware enabled solutions such as *Heartbeat* or through Microsoft Windows Server fail-over support. When manually switched, all that is involved is the pre-replication of the AgilePoint Server Config file.

3 Processing Modeling

Process models are defined by designers using Microsoft Visio-based AgilePoint Envision. Using the Envision extension for Microsoft Visio, users drag and drop Visio shapes (i.e., reusable IT assets in the form of process descriptions such as AgilePart, AgileWork, etc.) from the AgilePoint stencil onto the canvas and link them together. Alternatively, users can take an existing Visio model and adapt it for use within AgilePoint.

Either way, this approach allows users to quickly put a process model together. Process models may be saved in a Visio file format to facilitate sharing and collaboration throughout the design process. Once developed, the model is saved in an XML format (based on an extended version of XPD), and then deployed directly to the AgilePoint Server for execution.

More sophisticated AgilePoint enabled applications are built using the AgilePoint Developer application within Microsoft Visual Studio. This allows the developer to associate custom code with the process template, using an AgileStub. Moreover, developers can build libraries of AgileParts of pre-integrated Tasks and libraries of AgileWorks of dynamic manual activities that can adapt behaviors at runtime, according to meta data, placing them on a Visio Stencil to make them available to process modelers and business managers. As a result, the business analyst can reuse functionality established by the specialist IT developer, effectively enabling the business to leverage its existing IT assets.

The AgilePart and AgileWork functionality is very powerful. For example, each AgilePart can call specific functionality in an external application, yet provide it in a way that business analysts and users can deal with. For instance, AgileParts might be built as a generic Web Service call that, at runtime, prompts the user for the sort of service required. Web Services-based AgileParts provide an SOA (Service Oriented Architecture) development environment that enables the rapid creation of loosely coupled applications.

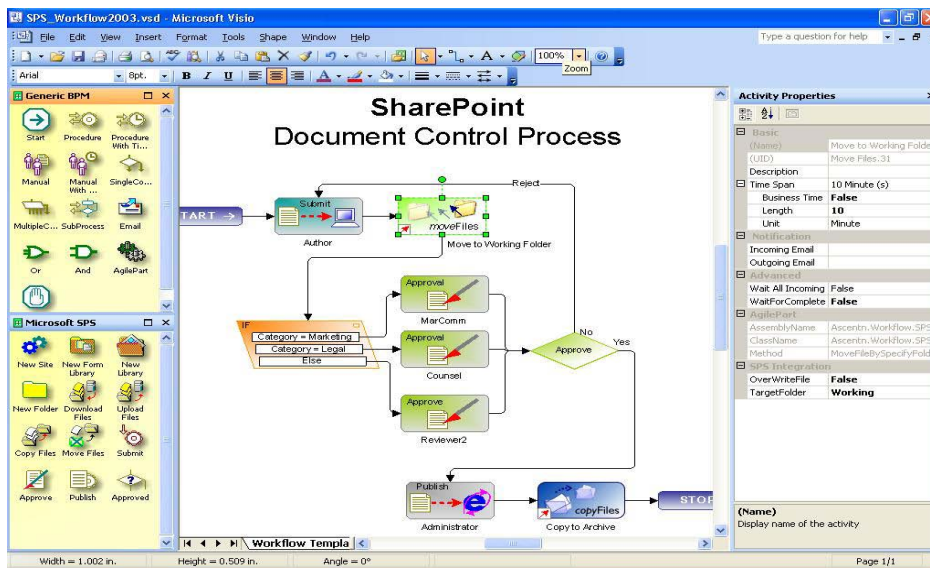


Figure 8. The AgilePoint modeling environment leverages Microsoft Visio; this example shows the SharePoint Server extended stencils.

Activities can trigger different email templates at different stages. Email templates contain dynamic

runtime data such as the activity's participant information or the unique process instance name.

3.1 Subprocesses

In effect, all subprocesses stand alone and are called dynamically at runtime. Any process model created in AgilePoint Envision can be loaded as a subprocess through the Subprocess activity. Both Web Services and native APIs are used as the mechanism to call subprocesses. Nested calls are supported.

The precise subprocess to use is either identified directly at design time, or determined at runtime via a process variable – in which case, the name of the subprocess is preceded by a \$ (\$SubProcess_Name). As a result, the subprocess could be determined by a Web Service call (performed by an AgilePart), a business rule, etc. Once a subprocess is called, an instance of the subprocess is created, allowing the developer to customize the subprocess instance programmatically at runtime for use within the parent process, without affecting the original subprocess template.

3.2 Shared Data Space

AgilePoint's Custom Attributes mechanism provides process and application related data management within a process instance or across multiple process instances. This data is used by the AgilePoint to perform specific data driven business logic, such as decision-making and conditional branching. Users can import or construct process and application related data through the Custom Process Property windows in the Envision component. Internally, AgilePoint uses an XML schema to capture and represent custom attributes.

When used with InfoPath, AgilePoint directly inherits the XML schema of the InfoPath form. The same capability is extended for Microsoft Office 2007 through OpenXML support. The process modeler merely creates the new process based on that form, immediately inheriting all the structure and definition therein. Individual fields are then available for use within the Activities of the model, such as the multi-condition business rule activity to drive process logic. Moreover, when used with InfoPath, the different views are also available to bind to the different Activities. When uploaded to the AgilePoint Server, the modeler can select to upload the form to SharePoint, and AgilePoint will automatically create the forms library if it does not already exist.

Custom Attributes can also store information relevant to a process instance, such as uploaded electronic documents, scanned images, or any other digital content. This information could be stored in an external database or content management system, etc. A unique identifier is stored with the AgilePoint process instance construct to facilitate retrieval.

3.3 Forms

AgilePoint advocates Microsoft-based forms technologies such as InfoPath, ASP.NET, WinForms, Excel forms, etc. As discussed above, it enables the firms to directly incorporate InfoPath forms into their processes and leverage InfoPath XML data schema.

Through AgilePoint process-aware Web Controls, AgilePoint extends the same facilities to standard ASP.NET forms. These process-aware AgilePoint Web Controls are based on the standard ASP.NET Web Controls and are accessible from within Visual Studio. Further, developers can import an ASP.NET application and quickly convert it to work with an AgilePoint enabled process. This feature can support a Lotus Notes to .NET migration (converting a Lotus Notes application to SharePoint and InfoPath, or ASP.NET and then utilizing AgilePoint to rebuild the process definition and support its routing).

Through AgilePoint Developer, additional application-specific functionality can be added to an AgilePoint Web Control (or in combination) to create reusable, composite user interface controls that can be shared across multiple forms. These composite user interface controls can be easily configured to implement an access control policy across users by controlling the visibility, the read/write permissions, etc. AgilePoint Developer can also be used to create Win Forms for the user interface. Developed

ASP.NET and Win Forms applications are exposed in Envision as a resource item to associate with process steps.

3.4 Time

AgilePoint developers specify the desired relative or absolute time for the completion of an activity, providing a benchmark for performance analysis and improvement. The process designer can also specify an absolute deadline for the complete process. When a *time out* occurs at runtime (at the activity or process level), the system invokes a predefined escalation action. Alternatively, escalation can be dynamically assigned to a user-definable exception handling module (implemented as an AgileConnector through the AgilePoint API).

3.5 Process Optimization and Simulation

A simulation facility is provided within the Envision tool to provide a pre-production simulation capability to help business identify potential bottlenecks and predict cost allocations. Having specified the percentage splits at decision points, and the time/cost allocations, the tool exercises the model helping the user identify bottlenecks. Additional simulation enhancements will be supported in later releases of the Envision tool.

4 Business Rules

Business rules within AgilePoint are reflected in the default visual business rule object. This extensible dialogue allows the process designer to reflect any conditions required. Alternatively, custom business rules are developed in Visual Studio to override the default business rule object, or a reusable custom business rule object is developed as an AgilePart.

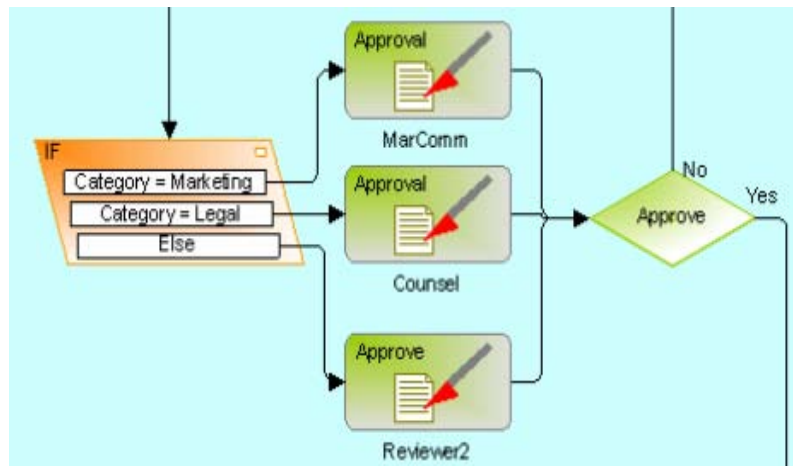


Figure 9. Visual business rule object.

AgilePoint can also integrate with a third party business rules engine through either Web Services or its API. For example, AgilePoint uses a generic Web Service AgilePart to access Microsoft BizTalk business rules engine. The configurable XML interface enables users to specify and bind process variables to the business rules engine.

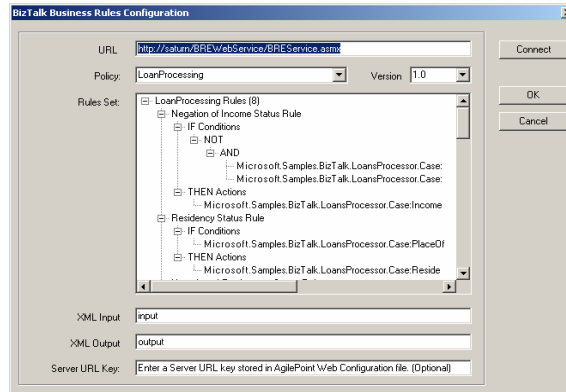


Figure 10. Calling the BizTalk Business Rules Engine

5 Integration

The primary mechanism for integration of third-party applications into AgilePoint processes is through AgilePoint Developer (Visual Studio). AgilePoint provides the *AgileStub* and *AgilePart* functionality within AgilePoint Developer to facilitate this.

AgileStubs associate external functionality with specific process templates. AgileStubs automatically create the necessary skeleton for including user developed assemblies and classes. Existing Visual Studio projects can be upgraded to include AgileStub functionality, enabling them to be accessed by AgilePoint process models. Once the necessary development has been completed, AgileStub is exposed to Envision to allow its association with a process step. The assemblies and classes need to be available on the AgilePoint Server before the process template is deployed.

Alternatively (and probably more commonly), AgileParts provide an integration mechanism to embed external functionality that is reusable across process templates. An AgilePart encapsulates any desired functionality into a reusable Activity object, which is then made available within Envision-based models (designers then just drag and drop them onto process models). Effectively, AgileParts deliver an object oriented capability at the process level, enabling the business to leverage its existing IT assets in a reusable fashion, accessible by business managers and process designers (without needing to know how the AgilePart works). AgileParts can encapsulate past (pre .NET) developments such as VB, COM, C++, or even Java, allowing them to be leveraged as .NET powered BPM applications through AgilePoint.

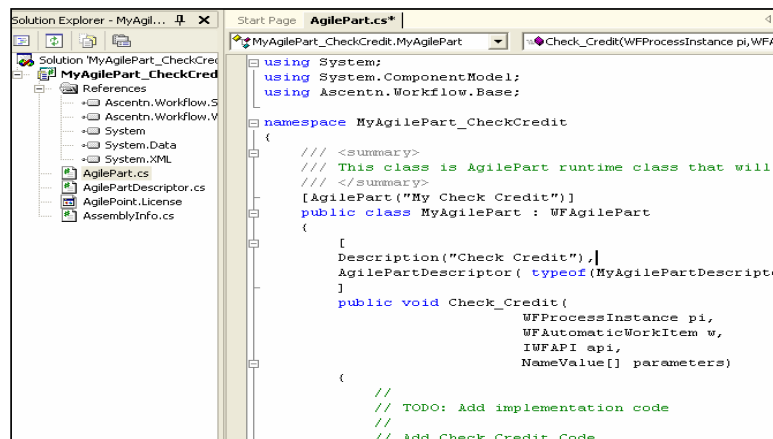


Figure 11. AgilePoint Developer automatically generates the necessary integration logic.

When embedding AgilePoint functionality into third party applications (i.e., the reverse of the above), developers use Visual Studio and call AgilePoint APIs for either server-side or client-side functionality.

Ascentn

AgilePoint, Ver. 3.2

APIs available include bindings for C#, VB.NET, C++, Web Services, Messaging Queues, COM, etc. When creating AgilePoint enabled Web applications, AgilePoint will also incorporate skeletons of process-related interfaces such as task management, process viewer, etc.

6 Organizational Structure

A pre-defined set of Roles (Administrators, Process Runtime Managers, Process Template Designers, and Users) is added into the system at installation. These are built-in roles that are used to provide basic access rights to different classes of users. Each of these built-in roles consists of a collection of access rights that can be further modified or customized. Additional Roles are added as deemed necessary.

When defining process models, the process designer either manually enters the Participant, or he selects an AgilePoint User, an AgilePoint Role, or an AgilePoint Group. Users can also select an Active Directory Group or select a user directly from the Active Directory or LDAP. The Participant can also be determined at runtime via a process variable – in which case, the name of the Participant is preceded by a \$, e.g., \$Participant_Name, which can be a user, role, or a group.

7 Process Adaptability

Unlike traditional code-generation based workflow and BPM tools turning static process models into rigid and monolithic executables that cannot adapt freely to changes at runtime, AgilePoint's XML based architecture provides a variety and high degree of process adaptability supports that are keys to deliver business agility. AgilePoint Server supports versioning of process models. As new instances of work are created, the Engine copies the process model to support the case in hand. As a result, any updated version of the model will not affect existing work in progress. Moreover, this approach facilitates changes to the process model supporting the case in hand, should that be needed during execution.

Users can adapt the process in-flight, if needed, interacting with the model either through the AgilePoint API or via AgilePoint Enterprise Manager. For example, with Enterprise Manager, a user could bypass a task without performing the work, or *roll back* an already performed step. Further, a case can be *suspended*, *resumed* or *cancelled*. AgilePoint also enables the creation of user definable delegation rules via AgilePoint Enterprise Manager. If need be, the actual process model itself underpinning the case may be modified (by suitably authorized users).

Depending on the design of the model, users might be prompted to select dynamically an appropriate subprocess to bind to the parent at runtime. Alternatively, this could be driven programmatically or via business rules based on external events.

8 Process Lifecycle

AgilePoint Envision supports the deployment to more than one target server environment. This enables support for various stages, including development, testing, staging, production 1, production 2, etc.

Moreover, a separate component (Process Template Management) in the BPM Engine is provided to manage the lifecycle of the process models. The process models are tracked throughout their life (created, released, checked-out, checked-in, and retired). Only a *released* process model can be executed while all previous versions are *retired*. System Administrators, or users with appropriate permission, have full control over the deployment and versioning (facilitated through AgilePoint Enterprise Manager). If needed, the active version can be rolled back to a previous version. Reporting mechanisms can also be created to compare the performance between different versions.

9 Monitoring, Measurement, and Management Information

AgilePoint Enterprise Manager provides a collaborative hub and digital dashboard throughout the

Ascentn

AgilePoint, Ver. 3.2

process lifecycle. It supports out-of-the-box support for audit, activity, and real-time tracking of cases. AgilePoint Enterprise Manager is a web-based AgilePoint application itself created in ASP.NET. AgilePoint Enterprise Manager functionality and interfaces can also be exposed through SharePoint as Webparts.

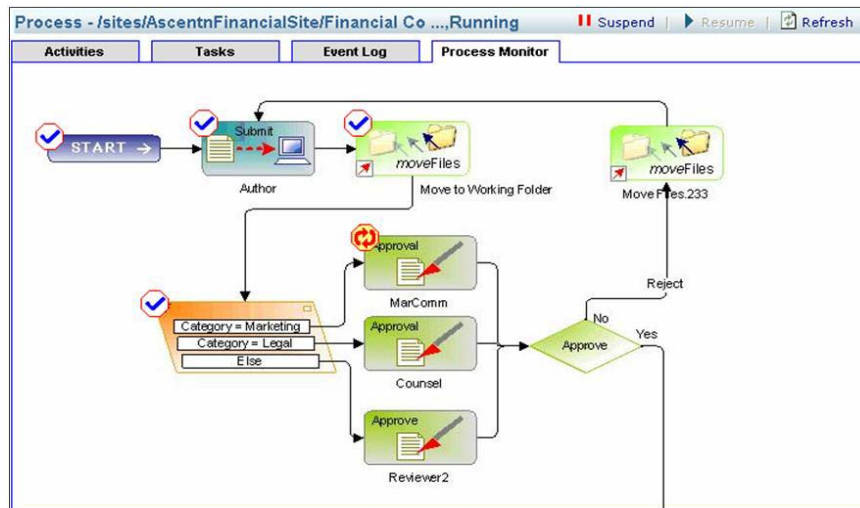


Figure 12. The AgilePoint Enterprise Manager provides a real-time monitoring window on the state of case.

AgilePoint lets users create user-configurable dashboards and reports and includes both real-time and after-the-fact reporting and analysis tools. It also provides information on the system status, server performance, application threading data, and monitoring.

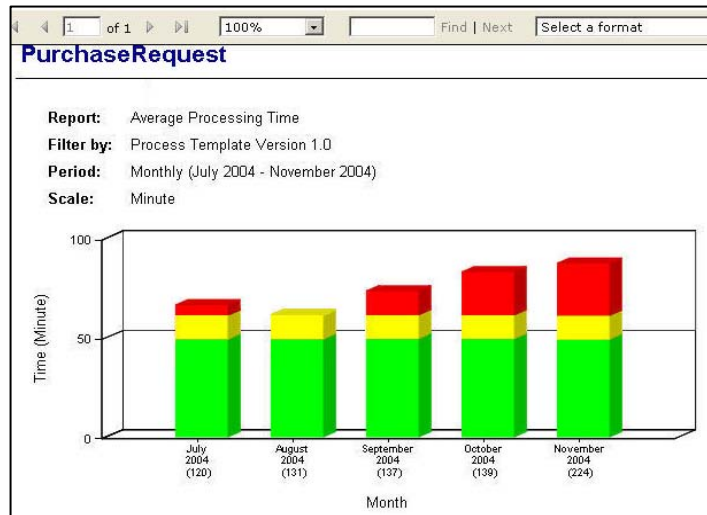


Figure 13. Integrate dashboards provide management with a view of work in the system and past performance.

10 Templates and Frameworks

AgilePoint provides a number of out-of-the-box templates—for example, a number of document management and control templates to leverage the functionality of Microsoft SharePoint Server are offered. The AgilePoint stencil for *Microsoft SPS* (SharePoint) are implemented as AgileParts. Together with SharePoint, AgilePoint enables businesses to leverage a wide variety of out-of-the-box InfoPath templates to deploy general business process management applications such as purchase authorization,

expense reports, etc., with virtually no code required.

AgilePoint OEMs are building vertical solutions to cover: Regulatory Compliance, Insurance, Document Management, Facilities Management, Financial, Legal, Life Science, etc.

11 Vendor

Ascentn was founded in 2002 with headquarters in Mountain View, California and offices in Japan, China, Taiwan, Spain, Germany, and a partner network throughout North and South America, Europe, Asia, and Middle East. At the time of writing, the company has nearly 100 end-user customers—with more than half multi-national companies including Global 2000's and close to a dozen OEM relationships. Ascentn has also observed a growing number of its customers converting their first project or departmental based AgilePoint implementation into enterprise-wide process foundation standard in as short a time as 6 months from their first implementation.

12 Cost

Driven by the company's value and usability objectives, Ascentn provides perhaps one of the lowest prices available in the industry, offering opportunities for significant price performance gain and ROIs. AgilePoint can be bought based on a fully configurable model. Out-of-the-box, AgilePoint provides 3 pre-configured editions – Starter, Professional, and Enterprise. Ascentn's goal is to enable any company to gradually adopt and practice BPM with minimal investment up front and to scale quickly with confidence and manageable incremental investments. For example, the Starter Edition, configured to support up to 20 state-less concurrent users (comparable to capacity for up to 500 named users), is under US \$10,000. Annual maintenance charges of 18% cover technical support and upgrades.