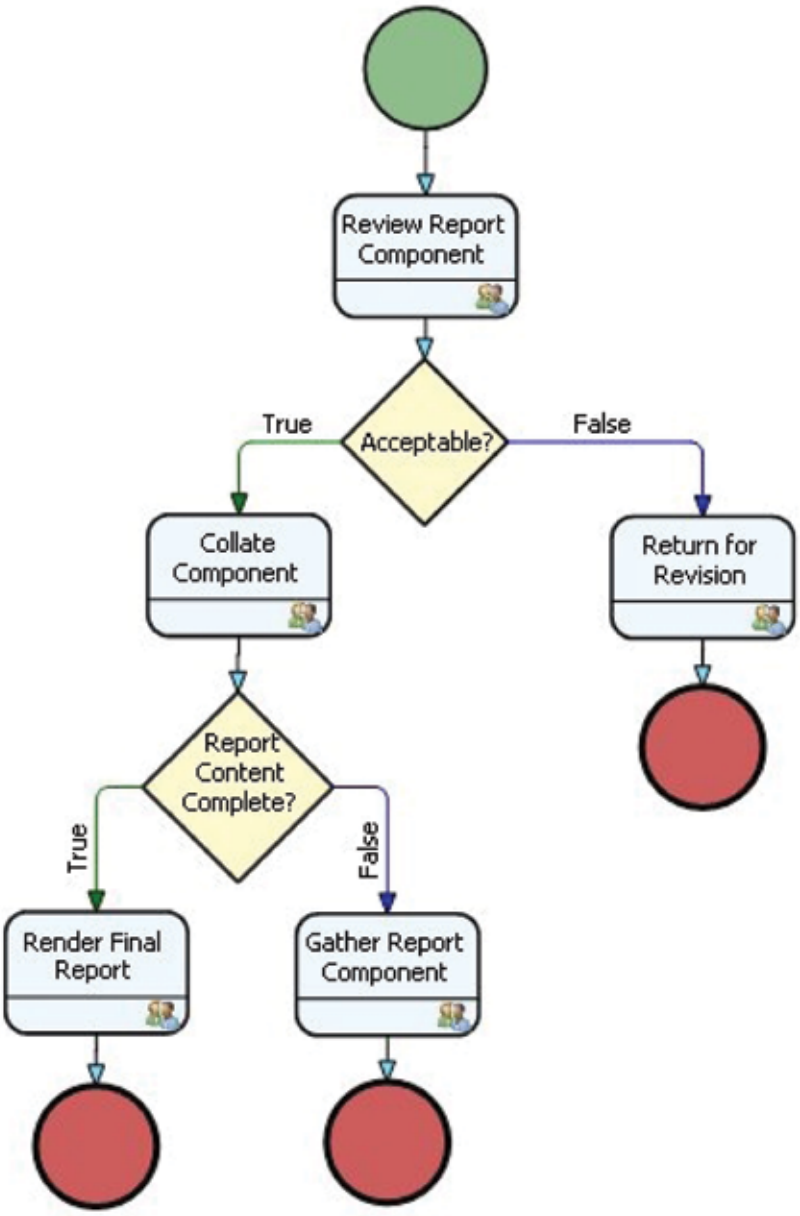
Singularity
PROCESS PLATFORM

TECHNICAL WHITE PAPER

A WHITE PAPER BY SINGULARITY

OCTOBER 2006



CONTENTS

PRODUCT & COMPANY OVERVIEW 3

SINGULARITY 3

THE SINGULARITY PROCESS PLATFORM 3

ARCHITECTURE 4

ARCHITECTURAL PRINCIPLES 4

Services Everywhere Architecture 4

True To Life 4

Easy 5

Self-optimizing 5

Technically Elegant 5

Zero Code 5

Built To Evolve 5

LOGICAL ARCHITECTURE 6

TECHNICAL ARCHITECTURE 7

Technical Architecture Schematic 7

PROCESS MODELING 8

MODELING PROCESSES 8

Handwriting and Shape Recognition 8

Electronic White Board Integration 9

XML 9

Enterprise Process Library 9

“Walkthrough” Functionality 9

Multiple Views of a Process 10

Business Process Modeling Notation (BPMN) Support 10

Data Access Wizard 10

Visio Integration 11

SUB PROCESSES 11

PROCESS VARIABLES 12

TIME & COST MANAGEMENT 12

PROCESS INTERFACE DESIGN 13

FORMS INTERFACE DESIGN 13

PROCESS PORTALS AND STANDARD INTERFACE LAYOUTS 13

SIMULATION 14

SINGULARITY’S RULES MANAGEMENT 14

THIRD PARTY RULE ENGINES 16

CONTENT, COLLABORATION & CASE 16

CONTENT MANAGEMENT 16

CASE MANAGEMENT 16

COLLABORATION 18

INTEGRATION 19

WEB SERVICES INTEGRATION 19

MICROSOFT INTEGRATION 19

PROCESS EXECUTION 20

WORKFLOW AND SERVICE ORCHESTRATION 20

ORGANIZATIONAL STRUCTURES AND DIRECTORIES 21

PROCESS ADAPTABILITY & DYNAMIC PROCESS INVOCATION 21

EVENT MANAGEMENT 22

EVENT TRIGGERS 22

EXCEPTION HANDLING 22

BAM, REPORTING AND MONITORING 23

DRAG-AND-DROP MONITORING OF KPIS 23

PERFORMANCE OPTIMIZATION 24

PROCESS CATALOGUE 25

CONFIGURATION MANAGEMENT 25

RELEASE MANAGEMENT 25

PILOTING 26

PROCESS SELF-DOCUMENTATION 26

SCALABILITY, AVAILABILITY, RELIABILITY 27

CONTACT INFORMATION 28

COPYRIGHT INFORMATION 28

PRODUCT & COMPANY OVERVIEW

SINGULARITY

Founded in 1994, Singularity is one of the largest and fastest-growing BPM vendors globally by revenue. After successfully developing integration and workflow products through the late 1990s, the Company launched our flagship Business Process Management suite, the Singularity Process Platform™, in 2000. Singularity focuses on providing process solutions to the Financial, Government and Telecommunication sectors and our customers include JPMorgan, the Bank of England, BT, BNP Paribas, the NHS, and over 100 central and regional Governments. A private company, Singularity is headquartered in Ireland, with employees located in offices in New York, London and Singapore and at our subsidiary in Hyderabad, India.

THE SINGULARITY PROCESS PLATFORM

The Singularity Process Platform (SPP or "the Platform") supports the entire life-cycle of business process management, from initial modeling, through execution, monitoring and optimization. Singularity differentiates itself through our focus on:

1. High speed collaborative process improvement
2. Early visualization of results
3. Rapid and iterative process deployment

Originally designed for the Microsoft environment, the Platform now also supports a Java J2EE process runtime engine. SPP exploits Microsoft .NET and COM+ technology along with open standards such as BPMN, Web Services and LDAP. The Platform integrates and builds on the broader Microsoft product range including Microsoft Office, SharePoint, SQL Server, Live Communication Server, Visual Studio, Outlook, Visio etc.

Singularity has adopted a Service Oriented Architecture for the Platform referred to as "Services Everywhere Architecture". In SPP, all components behave as a service, i.e. each subcomponent of the Platform can be called a service by a third-party component (call-in). Each service can call other services, i.e. each Platform component can extend, alter or substitute its service fulfillment by calling other services (call-out). Each service can modify its behavior by calling other services within SPP (self-call). Service orchestrations are themselves services – SPP orchestrates the sequencing of service provision into processes. All processes are available as services, and any service can be fulfilled by calling a process. This approach ensures support for the formal design goals of the Platform. These goals include the provision of an environment that is open, stateless, interoperable, multi-channel, multi-lingual, fault tolerant and scalable.

The process modeling environment supports easy-to-use point-and-click integration to external systems, with automated support for call outs to Web Services, and an easy-to-use data access wizard to guide business users through data integration steps. Work routing capabilities include fine-grained control navigating up or down organization hierarchies based on resource utilization, staff availability, skill set or priority. The incorporated simulation environment can utilize live and historical data, and different versions of processes simulated in parallel to compare relative costs and efficiencies, with output to Excel and other applications for subsequent analysis.

The Platform provides comprehensive Business Activity Monitoring that employs an "Event Manager" to gather Key Performance Indicator measurements during execution and then display this information in user-configurable dashboards. Further, in line with Singularity's focus on facilitating business ownership of business processes, a Forms Builder enables business users to design work queues and other process entry and display forms using simple drag and drop techniques. The Forms tool exposes

process related controls and data items, simplifying screen design and, as a result, reducing deployment times. The Platform also provides sophisticated version control and release management capabilities. A process catalogue enables the sharing and re-use of processes.

To support customers in the implementation, maintenance and evolution of solutions based on the Singularity Process Platform, Singularity provides a full range of consulting, implementation, education and customer care programmes.

ARCHITECTURE

The Singularity Process Platform is a component based system, developed using a Service Oriented Architecture. The following section provides the guiding architectural principles, the four key conceptual components and a detailed technical description of the system.

ARCHITECTURAL PRINCIPLES

Services Everywhere Architecture

In the Singularity product architecture, every component is a service: service orchestrations are services, processes are services. A "service" is a repeatable, self-contained collection of action and information. Services can be found, selected and executed by whatever business process requires them.

A "service orchestration" is simply a collection of services knit together and executed for some purpose. A business process can itself be seen as a collection of services executed in a particular sequence/order. There is little, if any, difference between a "service orchestration" and a "process". A business process can itself be a service, can be an orchestration of other services and can be invoked by other processes. Technical audiences familiar with SOA concepts will understand the "service orchestration" reference since this is standard language in the SOA world. Business audiences are generally more familiar with the idea of "process" and the concept that a process is composed of sub-steps also known as services.

The power of making everything a service (the "Services Everywhere Architecture") rests not only in the reusability of these services (so that there are not multiple, different and hard to maintain versions of services), importantly, services are also bound together at execution time. In other words, we can alter the insides of an "Individual Credit Check" service, e.g. add a new rating agency to the list, and the changed service is immediately employed in all subsequent executions of a process. There is no massive compiled software code that holds captive within it a "Individual Credit Check" procedure that cannot be changed without issuance of a new version of the complete "application". This removes a lot of inertia – "service-oriented" businesses are much more flexible and can respond more quickly when change is needed.

True To Life

Singularity uses realistic representations to drive the process improvement cycle. Singularity's products are designed to be intuitive for the non-technical business person. For example, a call centre manager will "recognize" his operation when he looks at business models built using Singularity products. Moreover, when the call centre manager wants to do a "what if", he will see the model of his business respond as his operation would, he'll see bottlenecks build and then unblock, he'll see resources sitting idle while others are only half-used. The guiding principle here is that we develop our products to be engaging at an everyday rather than a conceptual-only level for our customers. Our product innovations (such as in the Builder's handwriting recognition, electronic whiteboard integration and the visualization of costs of process steps by node size) support this 'true to life' principle.

Easy

Singularity has specified that we want all of our products and solutions to be easy to use, adopt, deploy and scale. This is reflected in our current product's intuitive process modeling environment, the browser based interfaces provided for accessing user's work queues, the fast and straightforward installation and deployment processes etc. It is also reflected in our focus on scalability.

Self-optimizing

Singularity was the first vendor to focus so strongly on the concept of the "self-optimization" of processes, i.e. building process solutions that are adaptive to circumstances and that continuously identify opportunities to optimize their execution. This architectural principle has been reflected in the design focus on dynamic invocation of processes, and making processes state aware so they can provide innate adaptability.

Technically Elegant

The Singularity Process Platform exploits open standards and technologies to achieve interoperability and performance advantages. SPP implements a multi-tier, multi-channel, multi-lingual, stateless, fault tolerant, and end-to-end processing architecture.

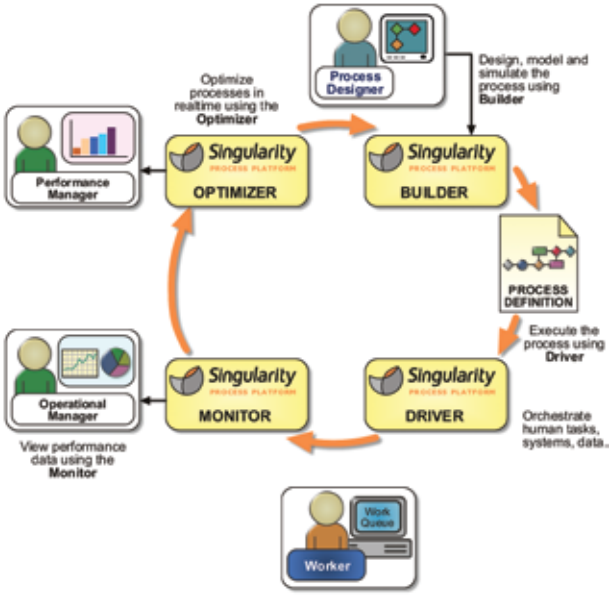
Zero Code

One of the motives for taking a process approach is the increasing emphasis on empowerment of business owners, and developing a balance between the needs of the business and the underlying dependence on the technology infrastructure. As part of achieving this empowerment, we have an explicit aim of achieving the bulk of business process integration and implementation without software coding – 'zero code'. This means the Singularity product auto-generates code where possible, e.g. for data integration (using a wizard) and for web-screen creation. As a result, the Platform boosts the productivity of both business and IT staff engaged in process improvement and management.

Built To Evolve

Process solutions should be built to support easy and frequent change. The Singularity solution is designed to enable the rapid design, deployment, analysis and redesign of processes in rapid iteration. It supports version control and release management, the provision of good cataloguing and library facilities, and provides strong support for deployment, administration and control of runtime processes.

LOGICAL ARCHITECTURE



Singularity's BPM suite is composed of four core logical components:

1. The Builder component provides an easy-to-use visual environment for modeling and testing processes using a point-and-click approach. The Builder can also import process maps created in Microsoft Visio or other modeling tools. Once modeling is complete the Builder enables customers to deploy and execute processes without additional coding or integration – the model is the process.

2. The Driver component is a robust and scalable process execution engine. Processes are executed according to the business logic, rules, policies and resource profiles specified in the map created using the Builder modeling component. The Driver guides process execution toward completion, orchestrating the combination of system and human activities. Reflecting the dynamic environment in which organizations operate, the Driver can react at runtime to changes in throughput, capacity, workload and utilization to modify its scheduling and routing of workload and process tasks. As exceptions occur, it can launch new processes to rectify the cause of the problem (without manual intervention).

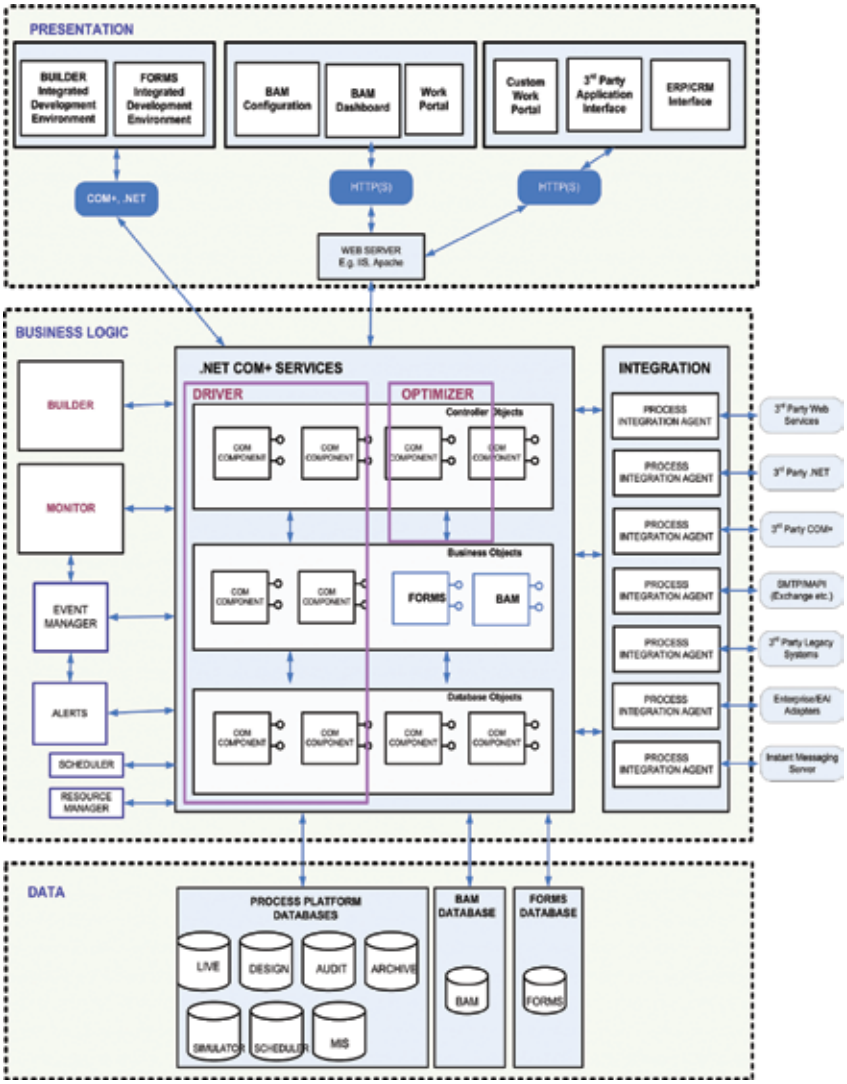
3. The Platform's Monitor provides comprehensive Business Activity Monitoring (BAM) functionality for viewing process performance. Organizations can define monitoring points across each step in a process life-cycle. Performance data and business intelligence is gathered at each of these points during process execution and reported in real-time. The data is also recorded and can be subsequently analyzed. The Monitor includes a dashboard for visual display of real-time and historic performance information. User definable reports are provided for viewing and analyzing historical data.

4. The Optimizer component enables execution-time improvement and self-optimization of business processes. It can execute a process differently, or execute a different process, depending on circumstances - for example during periods of increased demand. The Optimizer uses the performance data captured by the event-driven architecture of the Platform to determine the current state of the system and activate changes to running processes in real-time. The Optimizer can be configured to support tactical operational decisions (e.g. use a different process to get rid of a backlog) or to identify longer term process enhancements.

	Purpose	Services
Builder	Process modeling	Business Process Modeling Forms Design & Generation Simulation Resource Management Performance Management
Driver	Runtime process execution	Business Process Automation Workflow Exception and Escalation Management Event Management Real-time System to system integration Real-time Data integration
Monitor	Process monitoring and alerting	Business Activity Monitoring Real-time dash boarding Reporting Alerting Analytics
Optimizer	Process Optimization	Load balancing Dynamic Process invocation Real-time performance improvement

TECHNICAL ARCHITECTURE

Technical Architecture Schematic



Technically, the Singularity Process Platform has a modular, n-tier multi-threaded architecture. Each component implements a separate service, encapsulating a distinct aspect of the functionality. SPP comprises a set of stateless components, with all operational data and business relevant data managed within an enterprise database or multiple databases.

The Platform communicates with external systems through standard protocols such as COM+ or .NET. These external systems could include legacy applications, third party Web Services or other instances of the Singularity Process Platform.

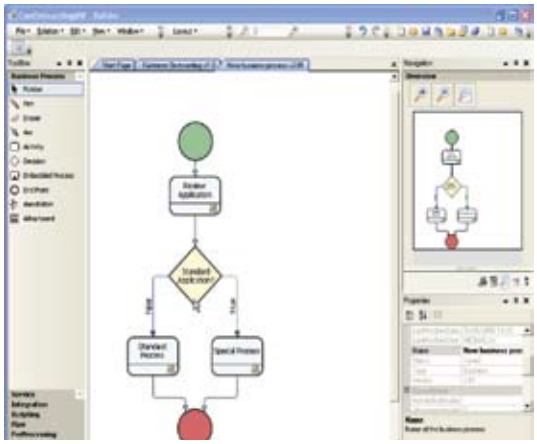
- The Platform's Driver is implemented through a series of stateless COM+ components hosted on .NET COM+ Services (which replace Microsoft Transactional Services as the Microsoft .NET application server). COM+ services provide common services such as transactional integrity, load-balancing, data access etc. Importantly, all of the Driver COM components must reside on the same .NET platform, although more than one Driver can be deployed.
- The Platform's Builder is implemented as a .NET service, i.e. it is not implemented as a COM+ component running on .NET COM+ Services, since it is not a runtime component.
- The Platform's Monitor is implemented as a .NET service. It is not implemented as a COM+ component running on .NET COM+ Services since it is not subject to the same runtime demands as the Driver components – generally it is polling for information at discrete intervals and then raising events through the Event Manager, and writing to a database. The Monitor can run on a separate platform to the Driver.
- The Platform's Optimizer is implemented through a combination of COM+ components running on COM+ Services. The Optimizer must run on the same platform as the Driver.

PROCESS MODELING

The Singularity Process Platform's design environment, the Builder, is a rich-client, graphical modeling tool that also supports multi-user access. Designers can analyze their models using the Simulator sub-component of the Builder. Screen designers use the Forms sub-component of the Builder, which is a web-screen design tool. Once satisfied with screen layout, clicking on the 'deploy' button automatically generates web-forms. SPP can also deliver information to users or applications using Web Services, COM+ and .NET Enterprise Services, across multiple channels including SMS, email and Instant Messaging.

MODELING PROCESSES

The Singularity Process Platform's Builder component provides a visual modeling environment for process designers and business analysts. The Builder can also import previously created process maps from Visio and other modeling tools. The modeling environment supports multi-user access and integrated process version and change control.



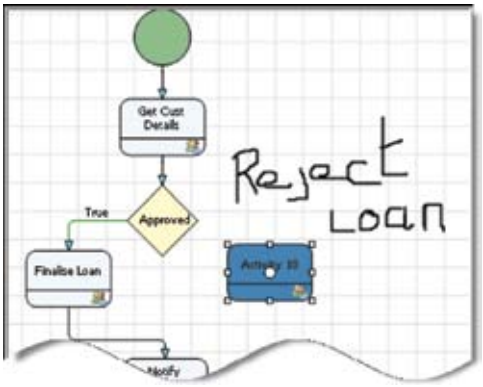
To create a process map, the designer opens a blank map, clicks on a palette to select an activity and then drops this onto a node on their map. An activity is an individual step (service) that forms part of the business process, e.g. get customer details. An activity in a process map can have various uses: a call to an automated script; a form to invoke; a call to a third party application; a request to a mail server to generate an e-mail, etc. The designer selects the type of activity that a node represents (a manual task, a Web Service, a scripted activity, an XML activity, a COM+ activity, etc.) and links it to others using a drag and click technique.

The designer specifies the resources (human and system) required for each activity node. This information supports the routing of the process instance based on factors such as cost, utilization, resource availability, staff-skill set or priority of the case in hand. The Platform also supports sophisticated routing needs via an Advanced Workflow Rules capability. For example, this feature might be used where: three out of a possible group of four are expected to review a document (either sequentially or in parallel); or, members of a group must be specifically excluded; or, the instigator of a process instance must be excluded from an approval task, etc.

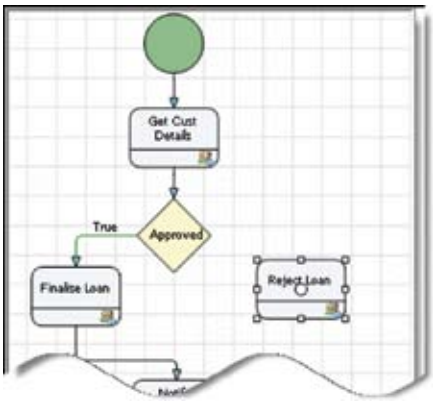
Using an integrated 'Web Services Tool', designers can discover Web Services on the Web and then register these within SPP environment. The expected parameters for the service are automatically identified from the WSDL description of the service. Using this tool, designers can build up a catalogue of Web Services, which are then available for use when designing process maps.

Handwriting and Shape Recognition

To facilitate design time collaboration on process modeling, Singularity has added shape and handwriting recognition to the Builder. This enables analysts to model processes in freehand with a pointing device during process workshops, without having to subsequently record the model – the Singularity Process Platform simultaneously builds up the corresponding map.



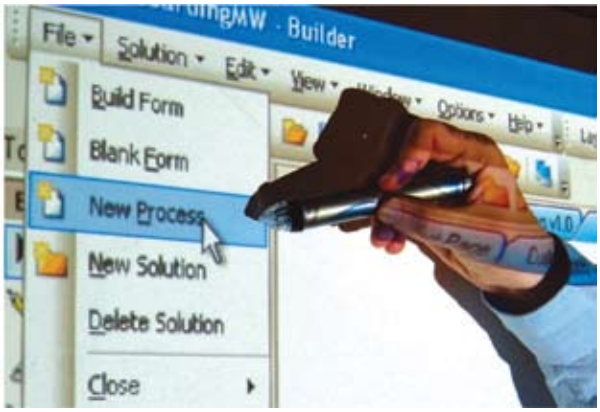
Before



After

Electronic White Board Integration

To further support design time collaboration on process modeling, the Singularity Process Platform supports the use of electronic white boards. Process designers can sketch models in freehand on a whiteboard, and the Singularity modeler simultaneously builds up the corresponding process map in its design environment.



This facilitates the rapid development of agreed process models while stakeholders are present, and avoids the usual delays between work-shopping a process and actually documenting and modeling it.

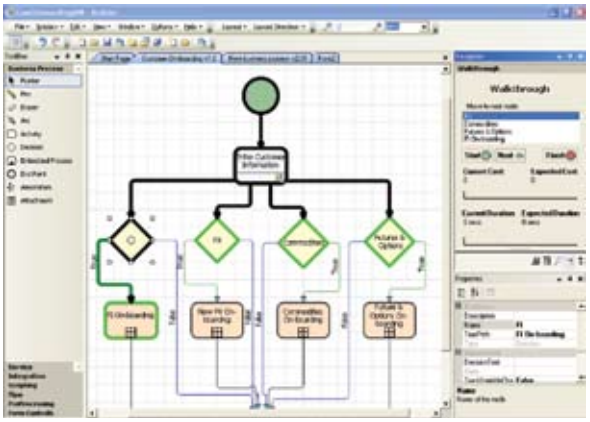
XML

SPP also includes comprehensive XML support through an XML parser (part of the standard XML node). As a result, SPP can manipulate XML documents directly without the need for 3rd party components. For example, SPP could identify the City from the XML address information returned in an online Zip code locator.

Enterprise Process Library

Configured activities, process variables, rules, resource definitions (and process maps) that are particularly useful can be stored in SPP's library for re-use in other process maps. Designers can drag-and-drop maps and activities from the library into their current maps. The process models themselves are stored in a proprietary XML format (relatively easily changed to another structure).

"Walkthrough" Functionality

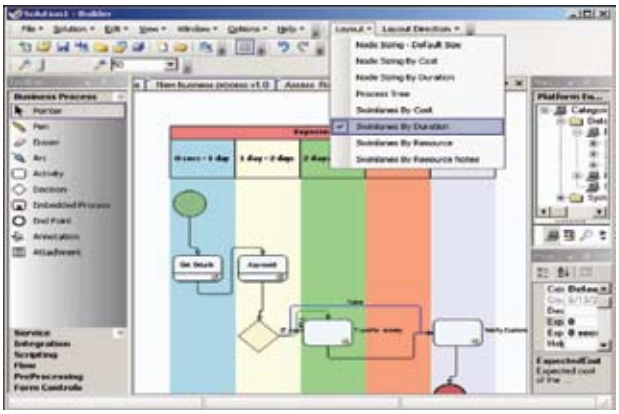


Walkthrough Functionality

SPP contains a "Walkthrough" feature that graphically steps through the paths in your process design, showing how different branches in the process will execute, how long the process will take, how much each process will cost to complete etc.

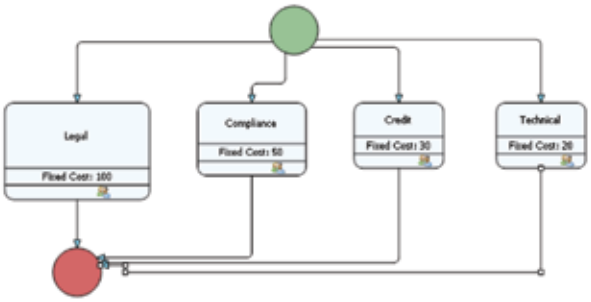
Multiple Views of a Process

Singularity provides a range of views onto a process map, letting analysts and business owners consider the likely impacts on their business of different design options, viewing the process from a number of angles. Swimlanes quickly illustrate resource or cost issues. For example, choosing to view the map using “Swimlanes by Duration” quickly highlights those parts of the process that absorb most time. You can also choose to view swimlanes by Cost and by Resource to gain similar insights.



Swimlane View

The Builder process design environment provides the optional ability to display map nodes based on their relative cost, or use of time or resources - using Node Sizing. This provides an immediate illustration of which parts of the process map expend most effort – for example, the more an activity costs the larger it is in the diagram.



Node Sizing

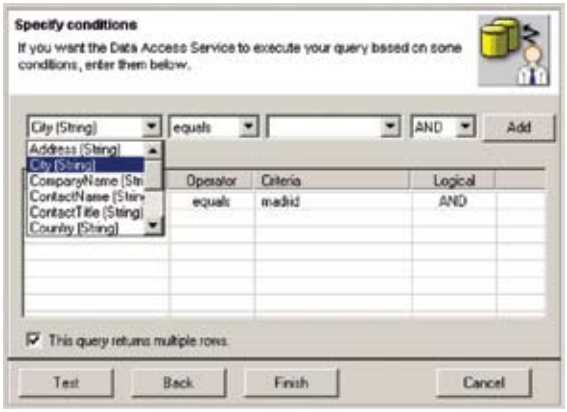
Business Process Modeling Notation (BPMN) Support

Singularity provides full Business Process Modeling Notation (BPMN) support. BPMN is a notation standard that specifies the icons and graphics to use when creating process maps. It was developed to provide a standard way of documenting processes which is easily understandable for both business users and technical developers. Previously there had been a variety of competing and proprietary notations and many firms ended up ‘reinventing the wheel’ by developing their own in-house notation. This made it difficult for different business analysts to interpret each others models, and to automate these models using different vendors’ modeling tools. BPMN is fast becoming the de facto standard for drawing business process models, making it easy to share and understand process models across and between organizations.

Data Access Wizard

The Data Access Services Wizard enables non-technical staff to access (i.e. read, write, update) a relational database without writing any code. It significantly reduces the technical knowledge required to model processes and data and allows Business Analysts and other non-technical staff to design their own process-data interactions.

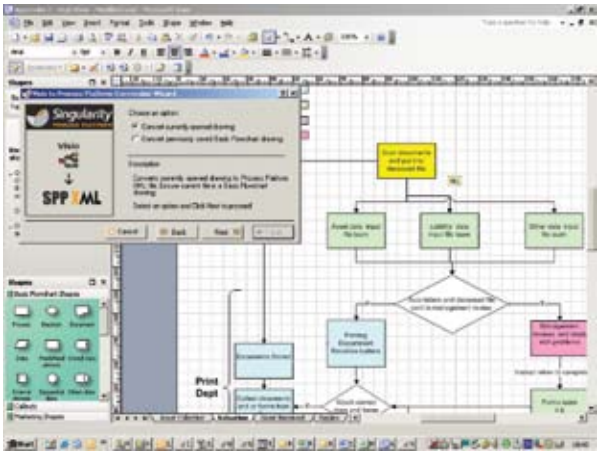
A Wizard interface helps process designers identify and select the data they require from the available databases and include this in the process step they are specifying, saving time while ensuring that the right data is accessed from the right source.



The wizard supports ordering of queries that return multiple rows, being able to call stored procedures, use aggregate functions and query against database views.

Visio Integration

The Visio Conversion Wizard enables process designers to convert Visio drawing charts into Singularity Process Platform (SPP) importable XML files. Business Process analysts can import their existing Visio flowcharts and other Business Process drawings into the Singularity Process Platform. The Platform can then implement these models ensuring that the business operates in accordance with the designed processes.



SUB PROCESSES

The Builder modeling environment provides a number of standard mechanisms to leverage sub process concepts. Sub processes are embedded at design time, or dynamically identified and called at runtime. The Platform supports both synchronous and asynchronous ("fire and forget") calling of sub processes.

"Nested" process maps are inserted as a single node into another map and are usually employed when the initial process map is getting too large, or if the designer has identified a process that may be re-used in other maps (as a first step toward supporting re-use). This may also apply in situations where work is carried out by another department or another company.

A "Sub Job Activity" calls a sub process as part of the parent process and exchanges input and output variables between the parent and sub-job. It also enables a number of instances of the sub-job to be initiated at the same time. Advanced Workflow Rules, mentioned above, can be applied to an entire sub-job.

A process can also be invoked at runtime – this is known as **dynamic process invocation**. This is the method used by the runtime Event Manager to respond to exception events (e.g. job time overrun), by invoking a related process to rectify the exception depending on the particular circumstances. The exception processes are defined at design time, but the selection of the appropriate exception process is made at runtime.

In the Builder, process designers can set a Transactional flag for Activities or sub-processes. This enables process roll-back in case of failure. Effectively this creates a two-phase commit capability to the database. When a sub-process has the transactional flag set, the entire sub-process becomes one atomic entity that must complete successfully.

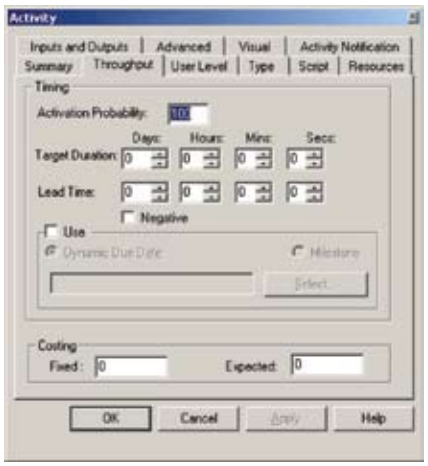
Finally, entire SPP processes may be published as Web Services and can subsequently be called by other processes.

PROCESS VARIABLES

Using the Builder, designers define local and global process variables. The scope of the variable can be limited to the current process, to the current and all child processes, or to the entire process space. These variables can include the declaration of arrays and complex variables. Designers create variables one at a time, which may then be associated with elements of an introspected Web Service. To facilitate cataloguing and library storage of potentially re-usable process components, process maps and variables can be allocated a "Category," which is used to store and retrieve associated re-usable components in the Singularity process library.

TIME & COST MANAGEMENT

SPP handles time at a number of different levels: during design-time specification of expected performance; during run-time monitoring of performance; and scheduling of running processes. This can apply at the level of an individual activity or for an entire process.



Setting expected duration for the an Activity

The Driver and Monitor track the actual elapsed time so that KPIs can be assessed. The Event Manager component within Driver traps standard exceptions such as process duration overrun, process cost overrun, activity duration overrun, activity overdue etc. Exceptions may initiate standard alert mechanisms (i.e. send email or SMS based notifications). Notifications delivery is itself based on a process (and therefore changeable). Process designers have full freedom to design customized processes that can be triggered by any event, exception or change of state.

As regards the timing of running processes, SPP includes an administrative interface to control how processes execute. The system uses an internal Business Calendar for specifying working and non-working days and standard working hours. It also includes a job-scheduling tool that allows administrators to set-up processes to run at regular intervals.

Cost can be associated with the activities in a process in the same manner. A cost figure can be specified for an activity, and also can be associated with a particular resource. These can be summed up to determine the expected cost of a process, and can subsequently be compared to the actual cost at any given point during process execution. These time and cost features enable support for Activity Based Costing. (Activity Based Costing is a method used to determine the cost of production of goods and services. It is a costing model that identifies the 'cost pools' or activity centers in an organization and assigns costs and services (cost drivers) based on the number of events or transactions involved in the process of providing a product or service).

PROCESS INTERFACE DESIGN

The Singularity Process Platform incorporates two sets of interface elements:

- Design time interfaces, used by process designers and business analysts when processes are being modeled and screens are being designed, as discussed above, and
- Runtime interfaces, used by process participants (workers) carrying out tasks and process managers who are monitoring process performance. These are discussed below

During runtime process execution, users interact with business processes via browser-based work queues, or browser-based portlets (known as web-parts in Microsoft's portal system Sharepoint). SPP fully integrates into Microsoft Office, so users can also interact with processes via Microsoft Outlook and other Office applications. The interfaces used are fully user-configurable.

Process managers use the Monitor's Business Activity Monitor (BAM) browser-based interface for monitoring the operational performance of processes, including a user-configurable dashboard.

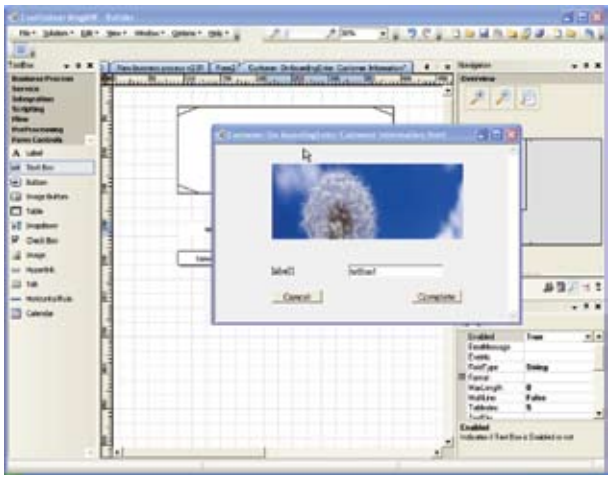
FORMS INTERFACE DESIGN

SPP provides a Forms sub-component as part of the Builder design-time modeling environment. The Forms designer is a wizard-driven tool, which gives designers the ability to quickly and easily create and deploy customizable screens, without requiring significant technical expertise or software coding. It provides an automatic screen generation facility, as well as the ability to customize forms.

The Forms Designer is tightly integrated to the underlying process maps. The Forms components can automatically pre-populate web screens with fields and process variables related to the underlying process map being modeled – it removes the major manual step of creating these fields and linking them to instance variables and database fields.

For example, a Business Analyst can generate a process solution with multiple work queue web-pages, new case (process instance) pages and all corresponding activity pages without resorting to code—i.e. he/she can select a business process map and automatically generate the required web pages.

The Forms tool provides 'plug-ins' (adapters) to enable the translation of third-party form definitions into a Singularity format. It is also possible to use third party forms packages, although these do not provide the process-specific supports available in the Forms Generator.



Forms screen design and generation

PROCESS PORTALS AND STANDARD INTERFACE LAYOUTS

Singularity provides a default Process Portal framework and standard look-and-feel for process interfaces, to enable operational staff and process workers to interact with a process solution once it is deployed. These browser-based standard interfaces are laid out in a standard portal format, made up of changeable 'portlets' and quadrants that can be configured and personalized for specific customer requirements. If preferred, users

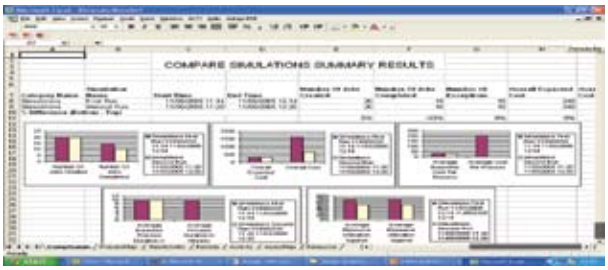
can ignore these standard layouts and use the Forms Designer to generate entirely customized screens suited to their particular requirements.

In addition to the process portal framework, Singularity provides a series of standard pre-built, solution specific interface layouts as part of Singularity's portfolio of Industry Process Accelerators. These solutions, built on top of the Singularity Process Platform for industry specific applications come with pre-configured web-screens that can be used for rapid deployment of working systems.

Example of Capital Markets interface

SIMULATION

The Simulator component of SPP's Builder equips Process Designers to simulate the performance of a process by setting up information on expected process throughput, resource costs, resource availability impact and so on. The Simulator supports "what-if?" analyzes, allows "competitive simulation" of different versions of a process in parallel, and gives design-time insight into likely future process performance. The Simulator returns test results showing idle resources, etc., and facilitates testing of exception handling and escalation processes, checking for adherence to Service Level Agreements. It can use both live and historical data when running process simulations. Output data can be displayed using Microsoft Excel, other Office products or more sophisticated analytical tools.



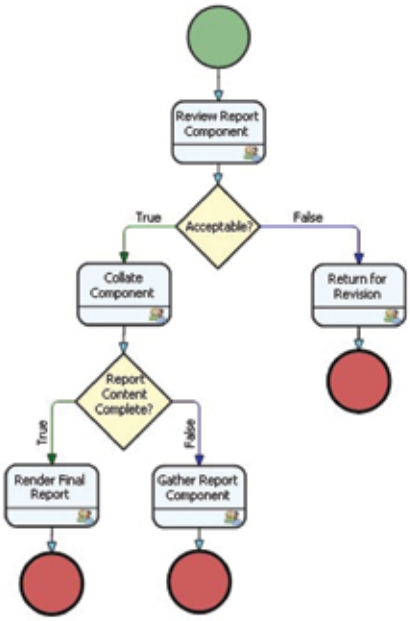
Sample Simulator output

SINGULARITY'S RULES MANAGEMENT

Business Rules define how a business should operate in given circumstances. For example, a rule could be "If the loan applicant is under 25, or earns less than $18,000, or has a poor credit rating then refuse the loan". Business Rules, in the context of processes, must be automatable and actionable.

Singularity provides support for Business Rules using a number of mechanisms. As the first step in business rules support, Singularity provides conditional routing – where a transition between steps in the process is based on the evaluation of a condition. If the condition evaluates to

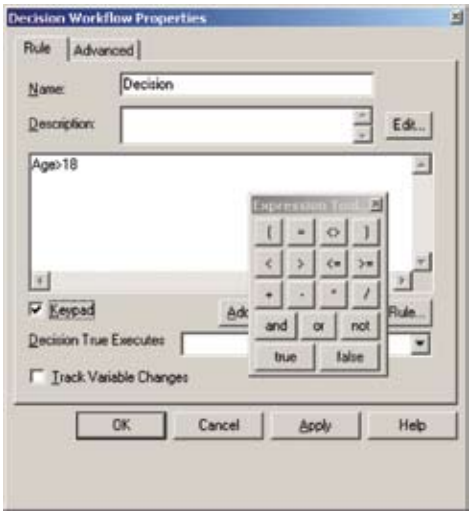
true, the process follows a particular path (often creating a fork in the process) – this is a conditional branch in a process map .



Process map with conditional branches

These conditional branches can be simple “true/false” evaluations of a single variable, but they can also route on more fine-grained decision logic.

A logical expression editor is provided that enables the analyst to define more complex business rules to be associated with a decision node within the process. The defined business rule executes using the variables of the process instance. This allows for complex decisions to be made based on process and system state.



Panel for creating Decisions

Rules are also reflected through the Advanced Workflow Rules concept. This facility supports highly sophisticated routing conditions that would otherwise require complex procedural models to resolve. Advanced Workflow Rules are used for specifying 3 types of conditions and 3 ways completing activities.

The three types of conditions are:

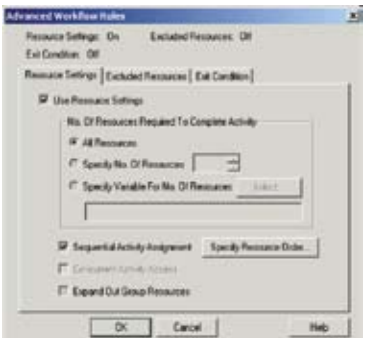
- Resource Settings: the resources and the order assignment (if any) in which the resources must complete the activity.
- Excluded Resources: any resources which can be excluded from completing the activity, e.g. a job’s creator.
- Exit Condition, e.g. a loan applicant is unemployed.

The three ways of controlling completion of the activities using advanced workflow rules are:

- Sequential activity assignment: permits selected resources to complete the activity in the order specified, i.e. sequentially.
- Concurrent activity access: permits specified resources to complete the activity in any order or at the same time.
- Non-concurrent activity assignment: permits required

resources to complete the activity in any order (this is not important), but only one resource can complete the activity at any one time.

Examples of this sort of advanced workflow rule functionality include routing work to three out of a possible group of four expected to review a document (either sequentially or in parallel); identifying members of a group that must be specifically excluded; or where the instigator of a process instance must be excluded from an approval task.



Panel for configuring Advanced Workflow Rules

THIRD PARTY RULE ENGINES

In some businesses, the rules are more complex than can be easily represented with conditional statements and process map conditional branches – for example, in medical insurance policy evaluation. Too many conditions, or too complex conditions, leave a business process map that is itself too complex and difficult to understand. Also, evaluating highly complex rule structures may impact system performance if the solution is not using optimal rule evaluation algorithms.

In such cases, Singularity uses third party rules engines for handling particularly demanding business rules performance requirements. Singularity's preferred rules partner is ILOG, one of the world's leading rules engine vendors, but we also integrate to other third party rules engines as required.

CONTENT, COLLABORATION & CASE

CONTENT MANAGEMENT

Singularity regularly deploys solutions in high-volume document driven environments, for example in government and banking engagements. To do this we provide integration to a number of document management systems so that processes can contain managed documents as attachments or by reference. When a document is attached, the Singularity solution can provide check-out, check-in, version control and other functions as required within processes, and transparent to the user. When a document is checked out to a process, Singularity stores it in an internal repository and then checks it back to the document management system when the process is complete.

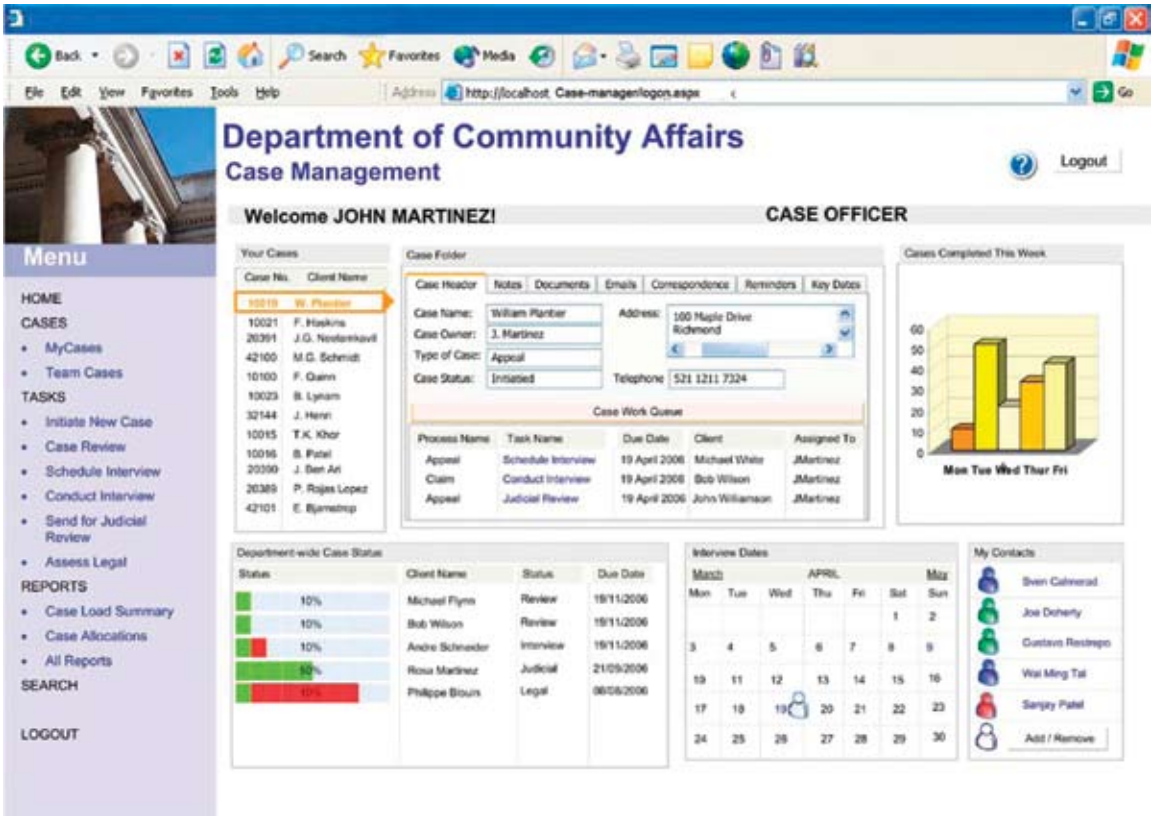
CASE MANAGEMENT

Singularity provides Case Management functionality in the Platform core product and additionally in the "Case Management" Accelerator framework. Case management is characterized as the management of the interactions between an individual and a company or government agency - for example, the handling of complaints at a government department, the handling of immigration applications or the management of planning applications and subsequent refusals and appeals.

Case Management has characteristics that separate it from other types or 'patterns' of process-based solutions. These characteristics include the need to handle unplanned 'out-of-bounds' breaks in process steps caused by external intervention, the need to manage and record large amounts of correspondence on a case that includes phone calls, emails and standard mail, the ability to link case processes to other cases, to have processes

dynamically invoke sub-process fragments in real-time and so on.

Perhaps the most important requirement is interface related - providing a consolidated real-time view of cases and all associated data, files and correspondence, to simplify the potentially overwhelming variety of data that can be associated with cases.

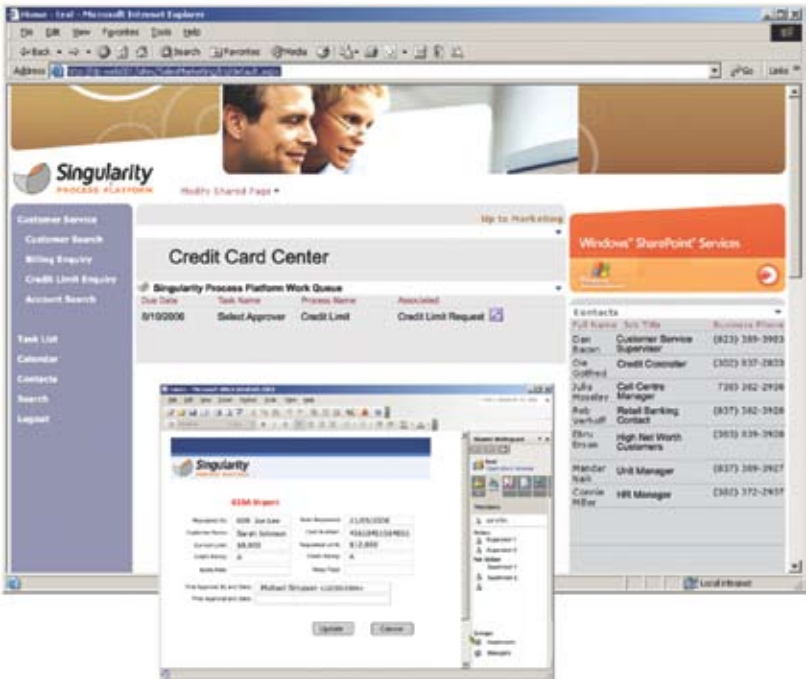


Example Case Management Interface

Singularity supports Case Management through Case Milestone Tracking for forward and backward case scheduling, through its Event Manager capability for monitoring and triggering case related events; through runtime selection and dynamic invocation of sub-processes; through state management features for supporting ad-hoc process breaks; through a correspondence handling component; and through default template interfaces. For example, the system can wait for the arrival of incoming documents, such as a planning application, from a document scanner (an event) and initiate new processes as soon as these documents are received.

COLLABORATION

The Singularity Process Platform provides extensive support for collaboration both at process design time and at execution time. At design time, analysts can use the integrated whiteboard capability to enable real-time collaboration among stakeholders during process modeling. The product, with its strong version control, supports process components being worked on by teams in parallel and subsequently recombined into a single process model. At runtime, the system drives work among teams, supporting complex routing and synchronizing manual activities. At the simplest level, tasks can be allocated to groups as well as individuals enabling a team to work on completion of a set of activities. As discussed above in relation to Content and Case Management, the Singularity Process Platform provides support for collaborative document review, as well as support for ad-hoc workflow to enable teams to collaborate on design documents and projects.



In addition, Singularity has uniquely integrated its BPM system to Microsoft's Live Communications Server so that teams can collaborate on time-critical work, share tasks, cooperatively develop documents and expedite urgent cases. This integration enables the Singularity Process Platform to detect colleagues' presence status – whether they are online, away from their desk, out-of-office etc. in order to more effectively route tasks among teams. Seamless integration enables SPP to support instant messaging as a means of collaboration among work colleagues during process flow e.g. to highlight an opportunity to conduct a high-value trade to fellow securities traders or raise an urgent issue related to a case.

INTEGRATION

The primary integration mechanisms for SPP are Web Services, COM+, .NET and scripting, using predefined Activity types. These Activity types (which appear as nodes on a process map) are used to model integration to and from a process. Internally, these nodes on a map are implemented through corresponding Process Integration Agents that support the particular interface (see Technical Architecture diagram in Page 7). Out-of-the-box, SPP can integrate to any system that supports any of these interfaces.

SPP supports integration by providing a variety of activity types in a process model, each of which supports a particular interface type. For example, a COM+ activity might integrate to the published API of a third party application. Similarly, a Web Service activity embeds the WSDL description of the target Web Service that provides the required methods. Developers can build up a library of integration components and re-use them directly in process models. Similarly, a catalogue of Web Services interfaces can be developed.

External applications that need to call SPP services (or its processes) can use the same approach. SPP exposes all of its functionality via Web Services, .NET and COM+ (there is also extensive API documentation).

In specific cases, Singularity has provided pre-built integration components to vertical-specific applications. For example, Singularity provides a connector for financial securities messaging applications used in Capital Markets, and connectors for major CRM systems in the Government sector.

WEB SERVICES INTEGRATION

SPP has in-built support for Web Services, including an online discovery facility. It also provides comprehensive support for XML, including a small XML parser. An example of business use of this capability would be including a Web Service call to a credit card authorization company, or the retrieval of delivery information on a dispatched order from a transportation partner. The XML parsing facility identifies the pertinent information, which is then passed to the invoking activity within the process.

MICROSOFT INTEGRATION

Singularity integrates into a broad range of Microsoft's product sets, including the Microsoft Office Suite. For example, Outlook can be used for sending email alerts and Excel is used for displaying Simulator output. SPP integrates to Visio and supports the import of Visio process maps into the Builder. SPP also supports direct interoperations with SharePoint Server and BizTalk Server, and operates with any ODBC-compliant database, including Microsoft SQL Server. Finally, Singularity has uniquely integrated its Business Process Management platform with Microsoft's Live Communication Server to support 'time critical processes' that exploit presence-aware features and enable both design time and runtime use of Instant Messaging and "presence detection".

PROCESS EXECUTION

The Driver component is the reliable and scalable process execution engine for the Singularity Process Platform. Processes are executed according to the business logic, rules, policies and resource profiles specified in the map created using the Builder component.

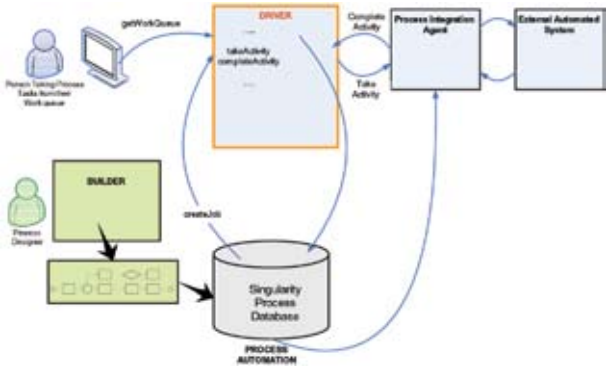
The component drives process execution to completion, orchestrating the combination of system and human activities. The Driver connects and coordinates the activity of applications, turning legacy systems into reusable assets that can easily be combined in new, valuable ways.

Reflecting the dynamic environment in which organizations operate, the Driver can react at runtime to changes in throughput, capacity, workload and utilization to modify its scheduling and routing of workload and process tasks. Should an exception occur, it can launch a new 'Escalation' process to rectify the cause of the problem without manual intervention.

WORKFLOW AND SERVICE ORCHESTRATION

The Singularity Process Platform and its Driver process engine uses a pure data-driven process execution architecture to drive workflow and orchestrate services. This architecture is faster, more scalable and more efficient than any competitor architecture and is completely transactional.

When a process analyst finishes modeling a process using the Builder design component, the resulting process map is saved to the Platform's database. The Platform represents all process models internally as data structures on the platform database - they can be exported in an XML format (SPP-XML) to other systems.



When a new process is initiated by a client (a human or external system), the Driver component creates an instance of the process for this client, which it then uses to control and coordinate subsequent process steps.

The execution of the process instance involves a repeated iteration of 'read next activity, evaluate, execute, read next activity'. At evaluation, the Driver component is asking "What is supposed to happen next?". The Driver examines the particular node of the process instance being executed to determine if it is a system or manual task, what dependents there are, and what advanced workflow rules need to be fired etc.

For example, if a the Driver determines that a node in the process instance being executed is a manual step, the Driver places this step into the part of the database that holds the client's work queues and assigns the step to the appropriate resource or group of resources (so it will appear on an individual's or group's work-queues). When an individual accesses his work-queue via the browser a "get work queue" request is sent to the Driver. The Driver reads the list of tasks from this individual's work-queue area and returns them to the via the browser. The person then selects an item from the work queue to work on, a 'take this activity' message is sent to the Driver and the Driver marks the activity as 'taken' for this process instance. When the staff member has completed the activity the Driver marks the activity as "Complete" and

the state of the process is changed. The Driver then reads the next node (activity) for this process instance.

If the Driver reads a decision node, it evaluates the decision and any corresponding logic to determine the 'true path' and thus which node to read next.

If the Driver reads an automated step, the processing is similar to that used for the manual step, with some small additions. Firstly, on reading and evaluating the step, the Driver does not assign any resources to the step (since it is not manual). Instead, it calls a Process Integration Agent and puts a task in the work-queue area for this agent (this area is known as the Thread Pool for automated activities). The Process Integration Agent, acting almost like a human client, sends a "take this activity" message to the Driver to indicate it will execute the task. It then executes the activity, by taking the inputs specified for the node, calling the relevant external system and passing those inputs as part of the call. It waits for the returned outputs from the external system and maps these to the process instance variables from the activity and sends the Driver an 'activity complete' message to indicate it is finished. Process Integration Agents can execute .NET, COM, web services, C#, VB.NET and other automated components, as well as invoking integration adaptors that link to standard enterprise systems.

ORGANIZATIONAL STRUCTURES AND DIRECTORIES

SPP supports the import of user data from Microsoft Active Directory, LDAP, or the Windows domain user-group structures. Alternatively, the organizational structure can be built directly within the product. There is strict security control over who can and cannot create resources and set their security levels in the modeling environment. It is also possible to model non-human resources enabling the routing of work to specific servers or applications.

Groups and Roles can be defined, which may also have supervisors and line managers associated with them. Against individual resources, the administrator can record information such as skill level, supervisor, security level, and cost per hour. This information is then available to support routing decisions and overall process cost calculations. For example, designers can specify that at a particular step in a process work should be routed to a particular group, to a staff member with particular skill set or security profile, or even specific individuals.

A rich set of configuration options are available to determine work routing. Depending on the process definition, the recipient might be the supervisor of the person who worked on a previous step (or the individual who created the job). Alternatively, a specific task can be delivered to all the members of a given role (sequentially or in parallel). Rules can be used to specify that if work is not completed by the original recipient within a given time, it will be redirected to a second or third person for completion (i.e. escalation of incomplete tasks up the organization hierarchy to the attention of supervisors and managers).

PROCESS ADAPTABILITY & DYNAMIC PROCESS INVOCATION

Singularity has developed significant process optimization capabilities in the product, supporting a significant degree of process adaptability at runtime. Anything from the priorities, due dates, resources of the process, or even structure of process can be dynamically modified by SPP at run-time. Resources can be swapped in or out, work can be routed differently, and sub-processes can be modified.

At run-time, every instance of a process map has a unique identifier, as does each activity within that instance, which the Singularity Process Platform's runtime engine uses for orchestration and tracking. If a process instance requires

modification, it is possible to change the underlying process model, effectively releasing a new version and then automatically linking that instance to run under the modified version of the process.

With skilled design, invoked sub-processes can be dynamically changed through user selection – i.e. workers can dynamically alter the current instance and all subsequent instances of that process. In addition, if a sub-process object is modified, any running process instance will by default use the latest version of the called sub process. With this sort of flexibility, it is straightforward to deploy a comprehensive and flexible case handling system.

The solution also supports dynamic process invocation, where one process is replaced by another, or a process is dynamically changed, to reflect current circumstances. A predictive analysis capability is also used, allowing SPP to spot repetitive patterns in the performance data. Together, these features support the self-optimization capability of the system, where SPP monitors the performance of the system and facilitates self-modification of running processes to enhance execution efficiency and ensure adherence to SLAs.

EVENT MANAGEMENT

The Singularity Process Platform implements both a Service Oriented Architecture and an Event Driven Architecture (EDA). Events are business or system incidents that can potentially initiate an action. Typically they are received by the SPP Driver component (i.e. process engine) as a message or a signal from a process or an external system. Singularity embodies its event capability in an Event Manager sub-component, which monitors for specific events and then responds to them in one of two ways – by launching a process (such as a 'compensating' process to correct an error event) or by completing an activity that has been in a wait state for the particular event.

EVENT TRIGGERS

The Singularity Process Platform both supports a predefined set of events which its Event Manager can monitor and allows analysts to specify their own key business and system events. Designers can add a node to a process map and specify that it should 'fire' an event as a trigger, to which the Event Manager responds at runtime by taking some action (generally, run another process, or 'wake' a waiting process instance). The Driver process engine persists the state of all process related data and instance variables in a database, and this data can also be used to trigger events.

EXCEPTION HANDLING

Exceptions can be system or business exceptions. System exceptions are errors raised during the execution of an automated step in a process, for example an external system that is called by the step may not be available. A business exception is an event issued by a user or system such as the cancellation of an order.

On detection of an exception event, the Driver suspends processing of the particular process instance and invokes

a special “exception handler” process flow – this exception flow will have been previously defined and associated with an activity, group of activities or a process as a whole. Singularity supports transactionality, where the work of an activity or group of activities can be “undone” if an exception occurs mid-execution.

SPP provides a number of out-of-the-box process templates and a comprehensive range of exception management processes for handling common exceptions such as Activity Overdue, Job-cost overrun and so on.

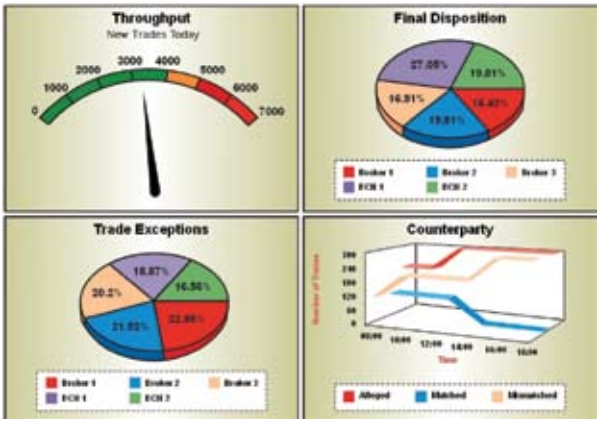
There are a variety of ‘out-of-the-box’ exception event handling process maps such as Delegation Process Maps, Emailing Process Maps, Management Information Update Process Maps and Process Map Distribution Processes.

The event-exception-compensation approach to escalation processes boosts worker productivity and supports continual process improvement. Singularity allows escalation management processes to be modeled, streamlined, automated, monitored and improved just like any other process. The result is that the number of exceptions can be successively reduced, improving worker productivity and business performance.

BAM, REPORTING AND MONITORING

The combination of events and process variables enables a rich set of functionality for process monitoring. Time-periods for individual activities are defined in the Builder component (Time-outs). Process designers can also set monitoring points at any step in a process, specifying the event flags raised. Performance and event data is gathered at run-time. The related business data is also captured and can be subsequently analyzed. All of these data are then available for reporting in real-time.

The browser-based Monitor component provides comprehensive Business Activity Monitoring (BAM) functionality for monitoring processes and viewing performance data. It includes a configurable dashboard and a set of user-definable reports to display real-time and historic performance information.



The browser-based dashboard

DRAG-AND-DROP MONITORING OF KPI’s

An Event design icon enables a process designer to simply and easily define a Business Activity Monitoring point in a process, by dragging and dropping an Event node onto the process map. The new node specifies what Key Performance Indicator will be updated by the process at that point in its execution e.g. “number of new

requests received in the last hour". This data is then reported on by the Business Activity Monitor.

SPP's Monitor can be easily configured to track a wide variety of business indicators, such as resource utilization, capacity availability, delivery cycle times, cost of execution, etc. It provides real-time performance visibility, including the throughput of systems and the productivity of process participants. This information can be used to identify process inefficiencies, performance bottlenecks and potential solutions.

The Monitor can be configured to trigger alerts or entirely new processes when pre-defined thresholds are reached. Designers configure an event flag that will trigger whenever there is an overrun of cost or time, at either the process or the activity level. For example, when a backlog of customer cases reaches a specific level or any other exception happens, the system can raise an alert or trigger a new process (to take corrective action).

The event manager is designed to support monitoring of both Singularity and third-party applications and services (rather than just process running within the Platform). It takes event feeds from external applications via Web Services, .NET or DB triggers or specially built 'gateways'. This ability to monitor all operational systems involved across a process allows a holistic view of business operations across the entire organization. The BAM component is available as an independent application from the SPP platform.

PERFORMANCE OPTIMIZATION

The SPP Optimizer component enables execution-time improvement and self-optimization of business processes. It can execute a process differently, or execute a different process, depending on circumstances - for example during periods of increased demand. Beyond simple load-balancing and optimized work routing, the Optimizer drives dynamic modifications to executing processes at runtime, based on the current state of the system. The Optimizer uses the performance data captured by the Event Manager to determine the current state of the system and activate changes to running processes in real-time. Anything from the priorities, due dates, process routing, or even structure of the process can be dynamically modified.

The Optimizer can be configured to support tactical operational decisions (e.g. "use a different process to get rid of a backlog today") or to identify longer-term process enhancements ("this process is repeatedly running into problems under these circumstances").

Examples of the Optimizers capabilities include:

- Performance-driven Optimization - reconfigures any aspect of a process to ensure achievement of pre-defined performance levels (such as cost or utilization thresholds).
- Revised Resource Allocation - the Platform supports automatic re-assignment and delegation of work in real-time, dependent on user-defined criteria such as resource utilization, resource availability, productivity and so on.
- Execution-time Process Optimization - the Optimizer enables organizations to automatically add staff and/or computing power or swap in replacement processes when a threshold is reached, for example if the backlog of customer orders reaches 6,000. Non-urgent steps in

a process can be temporarily by-passed during peak demand to ensure cases are dealt with promptly.

- Load-driven Process Prioritization - the Optimizer can automatically reallocate resources to high priority tasks and de-allocate from lower priority tasks at execution time when demand is heavy, returning to default resource allocations when demand returns to normal.

For Design time Optimization, the Builder includes a simulation sub-component, a 'wizard-driven' simulation and analysis environment that allows testing and 'what-if?' scenario analysis. Process Designers can simulate the workload of a process by setting up information on expected process throughput, resource costs, resource availability impact and so on. Simulator returns the test results showing idle resources, etc. Simulator facilitates testing of exception handling and escalation processes, checking for adherence to Service Level Agreements. It can use both live and historical data when running process simulations. The output data can then be displayed using Microsoft Excel, other Office products or more sophisticated analytical tools.

PROCESS CATALOGUE

SPP supports process version control, both at design time and during run-time. All design-time process maps have unique identifiers, and version numbers increment during both internal design updates and when released as a complete process map. SPP implements strict access control over who can modify and update process models. The Platform maintains all previous major and minor versions of a process model, and supports the deployment of both older and newer versions at the same time if desired.

Singularity provides a range of automated supports for version control, documentation, storage and safe distribution of processes, reflecting our view that processes should be managed as corporate assets.

CONFIGURATION MANAGEMENT

SPP includes a shared process library to support the sharing and distribution of commonly used processes. Among the capabilities provided to support "process asset management", SPP includes an automated facility to support the "safe sharing" of processes which "economizes," or strips out implementation specific details, allowing the structure of the process to be unchanged and reusable while protecting sensitive business information particular to a specific implementation of the process.

RELEASE MANAGEMENT

The Singularity Process Platform can trigger an event whenever an analyst, business staffer or IT staffer or other resource saves and releases a process map on which he is working. This event can itself be configured to trigger another process, e.g. a 'Process Release Management and Approval process' to ensure that a number of approvals have been gathered before the release of the new map could be completed. A wide range of process asset management procedures and policies

can be supported. For example routing a new version of a process model to the process owner before confirming formal release, or ensuring changes are not deployed into the live business environment without the proper authority. These features support the better management of a company's process assets, ensuring policy and procedure are complied with and supporting sophisticated governance of key business assets.

PILOTING

Singularity provides a 'pilot version control' capability, allowing two versions of the same process to run simultaneously. This supports the scenario where a modified version of a business process is piloted for a period in one department, while other employees continue to use the original.

PROCESS SELF-DOCUMENTATION

The Builder includes a Self Documentation facility that dynamically generates documentation on the process design. Depending on selections made by the designer, this can include general information about process map and detailed information about work queue definitions, resources, activities and associated parameters. The output is available in Microsoft Word, HTML or XML formats.

PLATFORMS

SPP supports the Windows Server 2000 and Windows Server 2003 platforms (Singularity is a Microsoft Gold Certified Partner) for its design time and runtime components. It also supports a J2EE version of its runtime process engine, the Driver.

Runtime clients and user interfaces are generally multi-platform, since they are accessed via a browser. Those design-time client components that are not browser-based support Windows XP and Windows 2000.

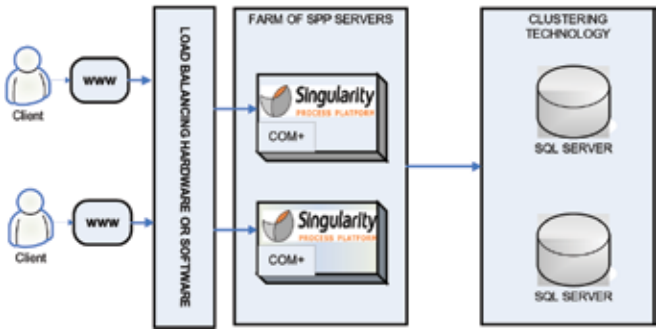
From a database point of view, SPP supports SQL-Server and Oracle. In principle any ODBC compliant database should suffice.

SCALABILITY, AVAILABILITY, RELIABILITY

Singularity has designed its solution with scalability in mind. Scalability has been considered from a number of views:

- Software component and infrastructure scalability
- Hardware scalability
- Platform scalability

SPP has a modular, stateless, multi-threaded architecture that allows for additional processing components to be added as demand increases. Specifically, the Platform adheres to a Service Oriented Architecture (SOA), with each 'module' implemented as a separate service, encapsulating a distinct aspect of the application's functionality. It is possible to increase the number of each of the 'modules' or services installed, limited only by hardware. Doubling the number of any given service doubles the whole system's capacity to handle that service's functionality. It also fully exploits the capabilities of Microsoft's .NET environment to support clustering, scalability, fail over and load balancing. Requests from the application to the database layer are mediated by server clustering technology to distribute accesses to the database servers. The database servers can operate in either active-active mode or active-passive mode.



It is possible to have a number of Driver components set up on multiple machines. Load balancing is handled either by hardware or Microsoft .NET Network Load Balancer (NLB), which passes the work to an available Driver. Microsoft Transaction Server (MTS) enforces transaction commit and rollback at the activity node within a process map.

SPP also supports the capability to set up a "Thread Pool" that can then be given priority over other work on the server. At design time, processes or specific steps are assigned to the Thread Pool providing the ability to ensure performance for specific parts of the process, while other, less important phases of the process execute more slowly.

Since SPP is a stateless system, the state for each process step is stored in and retrieved from a database. This increases the system's availability and reliability. If state was stored at the SPP server, when one server failed it would disrupt the processes of all connected clients. Instead, if an SPP component fails, the load balancer redirects the clients to the next available server, which picks up the state from the database layer.

CONTACT INFORMATION

Headquarters

100 Patrick Street

Derry

BT48 7EL

Northern Ireland

Tel + 44 28 7126 7767

Email: info@singularity.co.uk

Web: www.singularity.co.uk

Cable House, 4th Floor,

54-62 New Broad Street,

London ECM 1JJ

Tel: + 44 20 7862 4470

Fax: + 44 20 7826 4480

11 Penn Plaza,

5th Floor,

New York

NY 1001

Tel: + 1 212 946 2585

46 East Coast Road

#06-03 East Gate

Singapore 428766

Tel: + 65 9616 7732

COPYRIGHT INFORMATION

Singularity, the Singularity Process Platform and other company and product names referred to herein are trademarks or registered trademarks of Singularity Ltd and/or its subsidiaries or of the indicated companies.

COPYRIGHT NOTICE. No part of this publication may be reproduced, stored in a retrieval system or transmitted, in any form or by any means, photo- copying, recording or otherwise, without prior written consent of Singularity Ltd.

Copyright © 2006 Singularity Ltd. All rights reserved.