JIGSAW Final Report

Authors:

Jonas Hinn

Ekta Shah

Robin Kim

Annie Zhao

Zoe Ouyang

Master of Human-Computer Interaction

Carnegie Mellon University

Document History

Date	Person	Version	Change
July 30, 2007	Team	1.0	Initial Review by faculty
August 10, 2007	Team	2.0	Revised version for client review

Table of Contents

Table of Tables

Table of Figures

1 Executive Summary

Until recently, end users of business software found themselves frequently constrained by application boundaries when conducting their daily work. The functions and data they needed to accomplish routine tasks often resided in multiple systems, necessitating additional expenditures of effort and time. Thus, a gap is created between how business users work and what development organizations deliver. Advances in software development, however, now allow for functionality from existing applications to be more readily and easily combined in the creation of new composite applications that better integrate with real world business processes. Within a composite application landscape, parts from disparate business systems can be combined within a unified user interface that is tailored to the specific needs of users and based on the way they work. However, powerful composite application development tools alone do not guarantee a perfect fit with end user needs. Without meaningful user involvement at design time, business applications can become disconnected from the context within which they are used. As a result, the gap between user needs and application functionality can be perpetuated. For SAP to bridge this gap, it is important to create composite application design-time tools that allow business users to participate in a more meaningful manner in the development of their applications.

To further explore this approach, SAP teamed up with a group of students in the Masters of Human-Computer Interaction (HCI) program at Carnegie Mellon University (CMU) to investigate how to make SAP's NetWeaver design-time tools more usable for business users with minimal IT knowledge. This project spanned a period of nine (9) months and was divided into two phases. During Phase 1, we aimed to choose a design direction by conducting research to identify user needs. Our analysis included a usability evaluation of the NetWeaver design time tools, a literature review of various academic and scientific papers, a study of the competing products, user interviews and an extensive design ideation process. During Phase 2, we developed design concepts for user evaluation and iteratively built prototypes on validated concepts. This report details the iterative design process including the rationale behind our final design solution, thus marking the end of Phase 2.

In phase 1, we interviewed a broad spectrum of users ranging from technology-oriented to business users to understand their work practices and to gain insights related to business process design and implementation. In order to meet identified user needs, the team brainstormed design solutions for the concepts of documentation management, documentation generation, multi granularity views, rapid prototyping, collaboration, best practice communities, tracking project initiatives and simulation. The aggregate of these designs in the form of forty-one (41) concepts became the foundation of what would comprise an end-to-end solution.

After validating these concepts with user studies, we analyzed user feedback and in consultation with SAP, decided to focus on the concepts of multi granularity views, simulation and documentation management. The team created three rounds of paper prototypes that detailed these concepts and portrayed various use cases. After each round, we aggregated and analyzed the data gathered from user studies, and

refined our design according to user feedback. We found that business process experts first model the overall process before specifying the details and thus it would be important for our system to support this work behavior. Moreover, in addition to being able to efficiently communicating with different stakeholders with different needs, business process experts have the strong need to convince end-users to adapt to process changes.

After thoroughly testing our various designs, we proceeded to create a hi-fidelity interactive prototype using Microsoft Blend and Visual Studio. Our team was divided into two parts: designers who created the graphical components and developers who programmed the logic behind these components. The result of this collaboration is visible in the form of our interactive prototype called Jigsaw.

At its core, Jigsaw supports all stages of the business process modeling procedure within a unified system. The key advantage of our design is that it allows users to visually compose and configure their composite application, and this makes it very appealing for users with minimal IT know-how. Business process experts can model the overall process and then drill down to add a workflow to a process step. Once the workflow is built, users can test this workflow for efficiency and correctness. With respect to our original agenda, we believe that Jigsaw is well grounded in user intents and was robustly refined through multiple rounds of user testing.

2 Project Overview

SAP has been a provider of software solutions for businesses of various sizes for more than 30 years. An important product was the monolithic SAP R/3 Enterprise Resource Planning (ERP) software platform, which included modules for business domains such as Human Resources, Customer-Relation Management (CRM), Finance, Supply Chain Management, etc. Even though the leader in the ERP market in Western Europe [1], SAP faces strong competitors such as Oracle and IBM, which also offer complete ERP solutions in the same market. As a result, in order to stay as independent as possible from a single software vendor, enterprises use different products from a multitude of software vendors. These legacy enterprise applications often operate isolated from each other.

Composite Applications are used to integrate isolated enterprise applications to support companies' business processes. SAP NetWeaver Design-Time Tools are used to design and implement such Composite Applications. However, powerful composite application development tools alone do not guarantee a perfect fit with end user needs. Without meaningful user involvement at design time, business applications can become disconnected from the context within which they are used. Additionally, these tools are catered to software developers and are difficult to use by other stakeholders involved in business process design. Because of this, a gap was created between how business users worked and what development organizations delivered. Therefore, our goal was to create concepts and an interactive software prototype for business-oriented users who want to actively participate in Composite Application design.

3 Research Findings to Design Implications

During the first five months of our project time span, we applied various user research methods to discover opportunities for designing a business user enabled composite application design-time tool. We reviewed academic literature and business white paper, analyzed three other competing products, performed heuristic evaluation on the current tools, and conducted contextual inquiries with our target users.

Based on the research data, we identified that there is a great opportunity in designing a composite application design-time tool that integrates with the business process expert's current practice of business process modeling and testing. We also derived a set of design implications that drove and guided our final solution and prototype. The following sections summarize our findings. For a complete discussion, see our midterm report called “SAP Project-Midterm Report”.

3.1 Business Process Expert and Business Process Life-Cycle

During the course of our user studies and participant recruitment, we discovered a unique group of users who drive and facilitate the establishment of new or improved business processes in an enterprise setting. These people are often situated in the position comparable to an Information Technology (IT) department director or internal business consultant. Given their expertise in modeling and analyzing business processes coupled with a good understanding of what the system can and cannot do, we gave this role the title of a business process expert (BPX). We identified users with this role as our target user group as they are in the most favorable position where composite application design would take place.

Through interviewing with the Business Process Experts (BPXs), we were introduced to the idea of the business process lifecycle. According to the definition given by SAP, the business process lifecycle depicts an ongoing circle of phases and activities that are involved in “bringing a new business process to life, modifying an existing business process, and continually innovating a business process" [2]. The lifecycle contains fives phases as depicted in Figure 3-1.

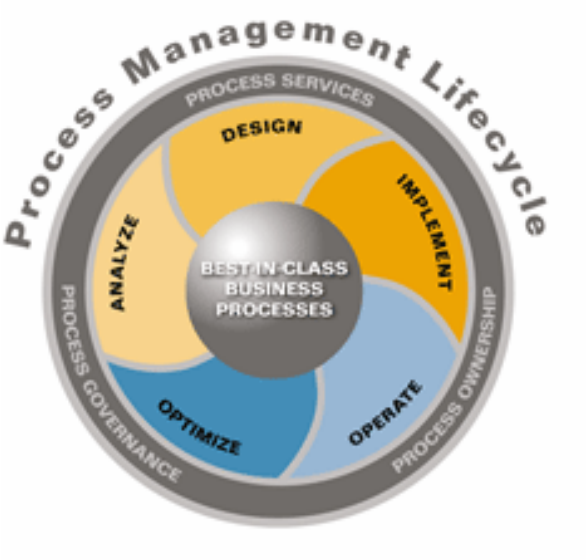


*Figure 3-1 Depiction of Business Process Lifecycle [**2**]*

Based on our research, the focus of the responsibilities of a BPX shift as a project moves forward to different phases in the business process lifecycle. These responsibilities in the different phases are summarized below:

- Analyze phase
  - Detecting and analyzing business issues and/or inefficiency in as-is process
  - Collecting and eliciting business requirements from end-users
- Design phase
  - Planning the to-be process based on defined needs and requirements
  - Gain buy-in for the process changes from the stakeholders (e.g. managers, end-users, etc.)
  - Designing and modeling the steps in detail of the to-be business process
- Implement phase
  - Working with the implementation team to define and document the functional specification and configuration details.
  - Creating and executing test scripts for the solution
- Operate phase
  - Arranging training materials and sessions
- Optimize phase
  - Monitoring and evaluating performance of the business process changes

Although composite application development does not happen until the later part of the design phase and implement phase, influences on the design of the composite are contributed through out the entire cycle. Meaningful involvement from business users may happen at multiple points in diverse forms. In particular, we considered

various opportunities around the themes of communication augmentation between business users and developers, intelligent documentation mechanism, collaborative development framework, and business-user friendly modeling environment.

3.2 Focused Design Opportunities

The business process lifecycle presents a myriad of design opportunities. In order to narrow our focus, we used Concept Validation to probe our users on the merits of various approaches to bridge the gap between business requirements and the composite applications that support them. For a complete list of our concepts and storyboards along with our results from concept validation, please refer to Appendix C.

We further pruned our design foci through a series of evaluative steps (see Section 4.3) and selected two primary areas to direct the design of our final solution:

- Enabling intuitive workflow modeling that leads to the creation of palpable visualizations
- Promoting business process changes and validating potential solutions before deployment

3.2.1 Enabling intuitive workflow modeling that leads to the creation of palpable visualizations

Business process modeling entails the communication of both high-level understanding and more granular information about implementation specification and logic. These areas are not mutually exclusive yet we observed a disconnect in switching between these two levels of detail in the work practices of business process modelers. In particular, SAP tool users often have to create the flow model with tools such as ARIS or Visio, and then find themselves redefining the model in Guided Procedure based on loose interpretations of the flow diagram. Such extra steps introduce not only inefficiency, but also the possibility of misconception when a different person, who usually turns out to be the developers, does the interpretation of the models. The frustration about these disparate interpretations was expressed both by business process experts and developers (see Consolidated Flow Model in Appendix A).

Our user research revealed that this problem has its origin in the diverged needs of two different audiences: the end-users and the developers. Therefore, we believe that enabling business process experts to model composite applications directly will enhance the expert's role as a broker between the end-users and the system they utilize. Such a modeling tool should allow the user to specify enough information for implementation. The tool should also produce a visualization of the process and workflow that business users can identify with.

3.2.2 Promoting business process changes and evaluating potential solutions

One of the key responsibilities of business process expert is to gain the stakeholders support on business process change. During our contextual inquiry sessions, the business process experts often expressed that the biggest challenge is in managing human dynamics and fostering adaptation of business process change. People have the tendency to resist changing their current workflow unless the benefit of change is made obvious. Hence, enormous amount of effort is focused on communicating the to-be business process to the stakeholders through various means of visualization,

prototypes, and comprehensive documentation. We believe that it is essential for the system to provide some solutions in alleviating the amount of extra work that has to be done for the business process expert to create convincing communicational deliverables.

3.3 Design Implications

Armed with the design foci, we reviewed our research findings and derived a set of design implications that served as the guidelines for the design of our system.

3.3.1 Leveraging commonly used notation and UI components

We consistently discovered that all actors in business process modeling use Visio or ARIS to create flow diagrams to visualize the high-level flow of a business process. It became clear that the people in the industry universally understood the standard flow diagram notations. In addition, the concept of swimlanes, that is used to specify who is working on a particular subset of the process, is highly utilized by the business process experts to visualize the responsibility hand-off between different roles and to spot bottlenecks in that arena. Therefore, we identified that it is important to employ flow diagram and the swimlane view as the main visualization of the business process and workflow. We also realized that leveraging the user interface controls and paradigms provided in the commonly used tools, such as Visio and Google Maps, might greatly reduce the learning curve of our system.

3.3.2 Varying granularity of process visualization

We observed from our contextual inquiry that business process experts often take the drill-down approach to understanding a business process. They would start looking at a business process from a high-level of abstraction in terms of big blocks of tasks and then drill down into each task to examine the sub-tasks. We identified three needs that can be extracted from such work behavior. The first need is to be able to see the big pictures of the entire business process. Without distractions from all the minor details, user can focus on the overall flow, major cut-off points, start and end conditions, and other high-level features of the process. The second need is to be able to focus on the details of individual sub-tasks and modules. According to the participants of our user studies, the complexity of the business process they deal with varies dramatically from process to process. In complicated processes, it is impossible, and often undesirable, to attend to all the details of an extensive process. However, this does not conflict with the third need in which users must be aware of the context of the sub task in the higher-level process.

It is often the case that the behavior of a sub process depends on the results that comes from previous sub processes. Insufficient data about the context and transition disorients users as they jump between the overall view and detailed view of business process visualization. This pitfall of the current SAP design-time tools was captured in our user research.

Our solution attempts to address the three needs mentioned above by introducing the paradigm of a zoomable user interface with varying granularity in the process visualization. In a zoomable user interface, user can zoom out to see the less detailed process overview or zoom in to manipulate the configuration and workflow of the sub-processes.

3.3.3 Seeing prototype in action

We learned from the user research that the most practical way of convincing end-users to accept and adapt to the changes in their work process is to present a working prototype of the future state of the process. Business process experts pointed out that business process flow diagram is insufficient to communicate the implications of process change to the end-users. Business process modeling notations were often found to be too abstract for the end-users to identify with (see the Consolidated Flow Model in Appendix A). More fundamentally, the end-users rarely care about other impacts and changes beyond the ones related to their own responsibility.

In addition, a prototype of the resulting composite application immediately relates to the end users' tasks. Our research data indicates that it is common practice for the business process expert to hold a session with representatives from the system users to present a proposed solution by walking through screenshots or drawings on the whiteboard (see the Consolidated Flow Model in Appendix A). Most participants we interviewed agreed that having an actual prototype would be tremendously valuable. However, creating an elaborated prototype is considered a tedious and time-consuming task given the current tools available. Hence, in our design we propose to allow users to quickly generate user interfaces for the end user based on the workflow model.

3.3.4 Providing contextual help

Developers and users of SAP systems complained about missing help functionality within tools. Additionally, according to research in the realm of end-user programming, knowing which operators and services logically flow together is one of the most difficult tasks for end-users in constructing a comprehensive workflow [3]. We aimed to incorporate the need of providing help based on the model's status and user's current task in the tool we designed. Solutions such as suggesting the next possible steps in a chain of actions would ease the user's burden in maintaining the syntactical integrity of the composite application model.

4 Final Design Solution

The design ideas and concepts found in user studies and the design process have led to an end-to-end solution in the form of a rich-client application, referred to as JIGSAW. It empowers business process experts with the functionality to model composite application supporting business processes and workflows. In addition, it provides an environment to immediately test workflows at design-time.

In this section, we first introduce the overall structure of JIGSAW and its design rationale. Then we present the main functionalities of our prototype by walking through a use case scenario followed by the design aspects.

4.1 JIGSAW Structure Overview

JIGSAW is a composite application modeling tool that integrates business process modeling, task-oriented workflow modeling, and prototype testing.

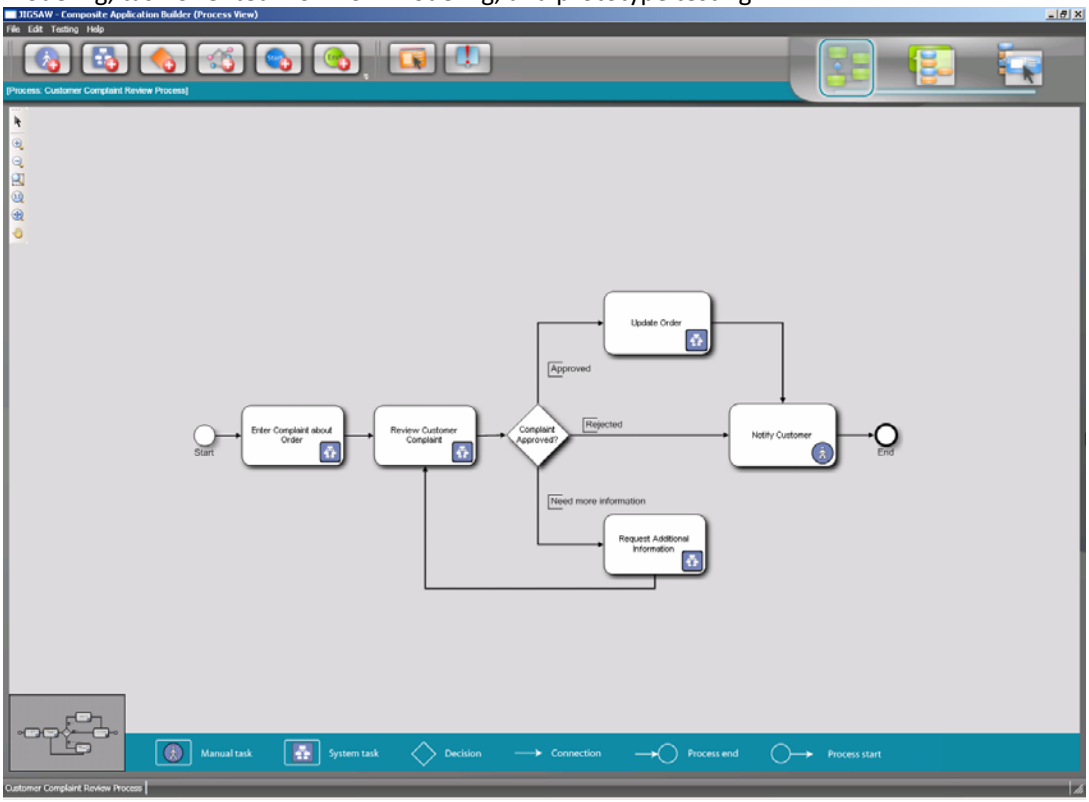


Figure 4-1. The business process modeling environment in JIGSAW.

At the top level, users can construct the high-level structure of a composite application in the flowcharting practice that is comparable to business process modeling (Figure 4-1). At the second level, after drilling down into a task in the process flow, users can manipulate the details of a task by changing the visual model of the workflow, such as adding an user interface screen or a service call and modifying data connections (Figure 4-2).

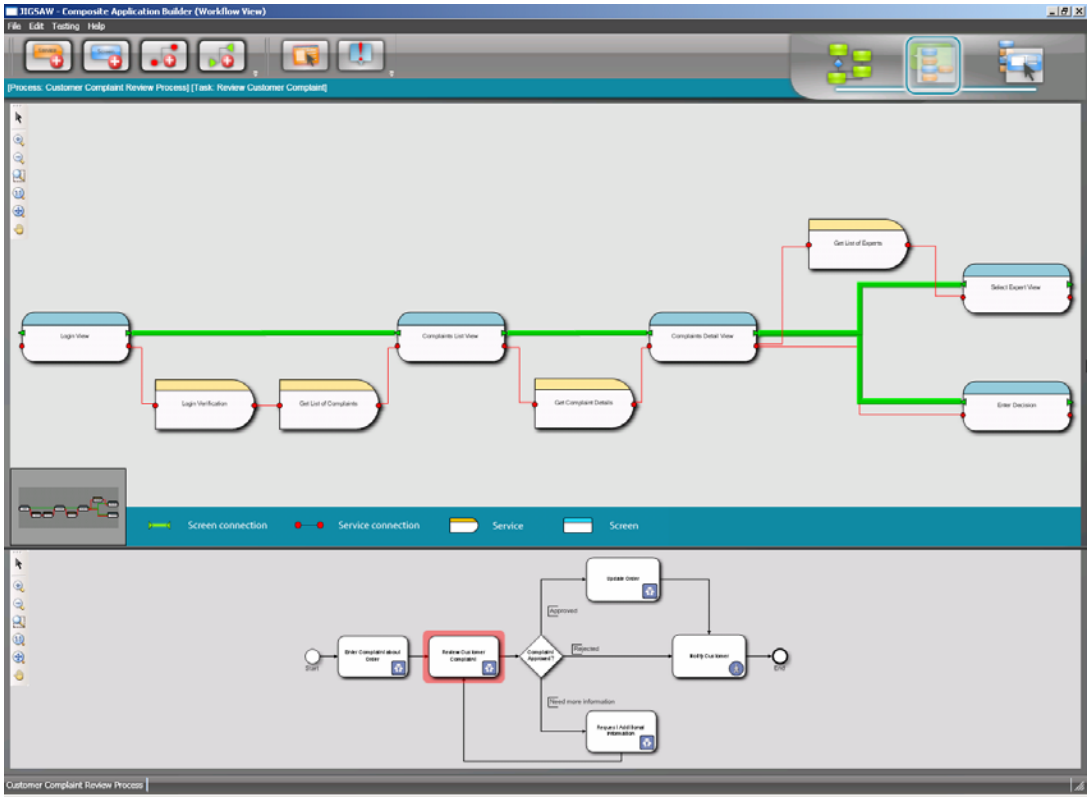


Figure 4-2. The workflow modeling environment in JIGSAW.

We decided to allow our target users to model composite application via the two-level visual programming approach for several reasons that are derived from our design opportunities and implications:

- The visual programming approach leverages business process experts' knowledge and proficiency in flowcharting derived from our design implication (Section 3.3.1).
- Allowing users to work directly on the visualization that is immediately presentable to business users reduces the amount of double-work in translating the composite application model as in the current practice (Section 3.2.1).
- Separate the visualization into two different levels not only supports the common top-down modeling practice (Section 3.3.2), it also allows users to focus on the different purposes each level of visualization fulfills. The process level visualization is meant to provide high-level abstraction that can be presented to the business users, whereas the workflow level modeling is meant to compose the functional specification that is required for defining a composite application (Section 3.2.1).

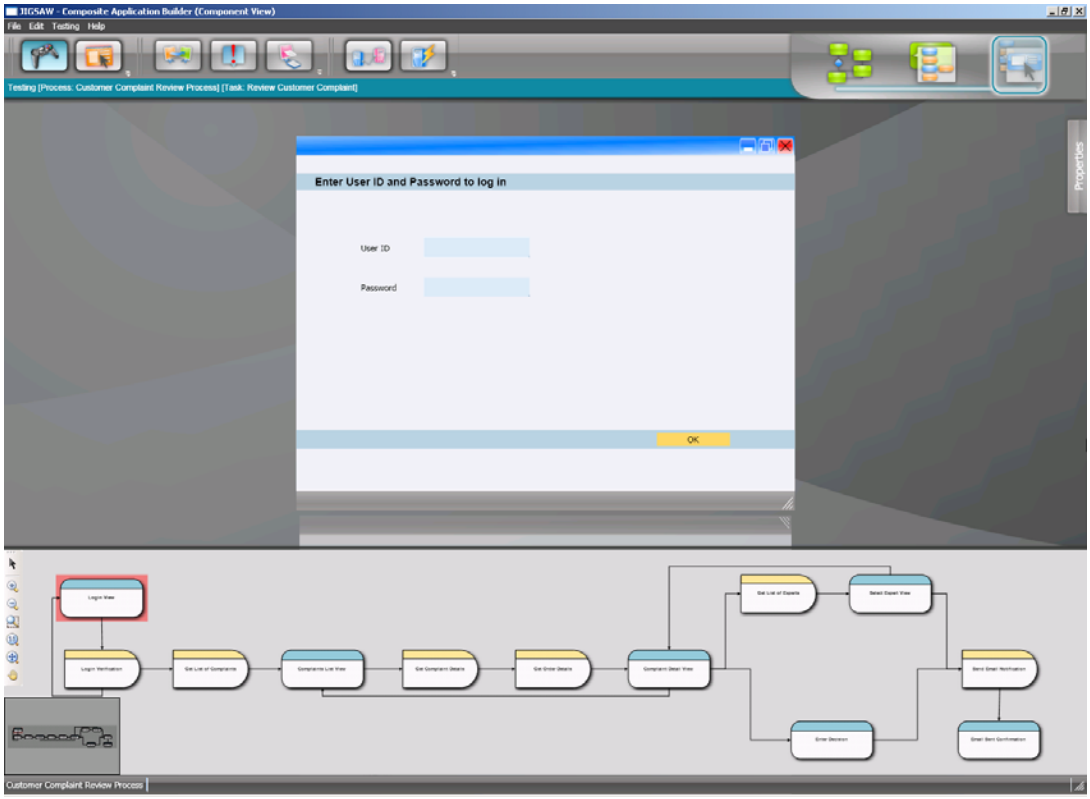


Figure 4-3. Testing environment in JIGSAW.

Once a workflow model is completed, users can test the resulting composite application by going into the testing environment (Figure 4-3). In the testing environment, a prototype of the composite is compiled from the workflow model. Users can interact with the composite and step through the entire workflow just like end-users would. Users can also inspect the input and output of the services executed along the workflow. At the end of testing, users can document the test details, including statistical data of the efficiency of the workflow.

During our user study, multiple business process designers expressed the importance and the pain of process testing, and the common wish for test automation (see Appendix E). Integrating the testing environment with the modeling environment strengthens our tool in a number of ways:

- Being able to quickly test the resulting composite application prototype helps the users to gain a better understanding of what they have modeled and how the end-users will interact with the composite. By providing the additional ability to inspect the data mapping between multiple components in the workflow, we assist the users to validate and debug the model against their design.
- A prototype is proven to be a powerful evaluation tool in assessing the efficiency and validity of the proposed business process. During concept validation, our participants expressed that one key value in process simulation is being able to statistically analyze the improvement of the process change (see Appendix E). Strong evidence of potential improvement often becomes the factor of determination in process change.
- The composite prototype is designed to be sharable with the end-users to serve the need of business process experts in presenting a concrete proposal to the end-users as discussed in our design implication (Section 3.3.3).

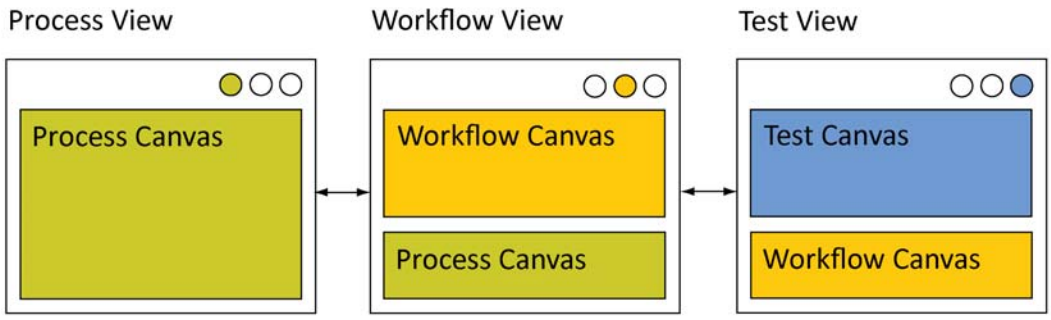


Figure 4-4. An abstraction of the relationship between three different environments. Notice that the Workflow View contains an overview of the Process Canvas and the Test View contains an overview of the Workflow Canvas.

The relationship between the three different environments is depicted in Figure 4-4. We deliberately separated the three different environments and provided a distinctive set of functionalities in each environment. We believed that the clear separation of functionality at different levels would support users' working pattern. The limited amount of functionality in each environment will also ease the mental workload while making decisions of what to work on.

In the workflow environment and testing environment, our system provides an overview of the level up, namely the process flow and the workflow respectively. This one-level-up overview serves as part of our solution in response to the user need of being aware of the context in the bigger process map while focusing on the details, which was identified as one of the design implications (Section 3.3.2). While working on modeling the details of the workflow, users can quickly refer to the process canvas to get information of the context. In the same manner, users can be aware of where they are in the workflow while testing individual screens or services in the testing environment.

4.2 Prototype Walkthrough

This section will walk the reader through several use cases of the JIGSAW interactive prototype. The walkthrough demonstrates how a business process expert can use JIGSAW to model a composite application supporting the customer complaint review process, without the need of extensive programming expertise. The walkthrough will reveals the key features of JIGSAW and their design aspects. Please refer to Section 5.3 for detailed specification of prototype functionalities.

4.2.1 Running the prototype

The reader is encouraged to run and experience the prototype while reading the walkthrough. Installation instruction of the prototype can be found in Section **Error! Reference source not found.**. To run the prototype, go the folder containing the prototype assemblies and execute “JIGSAWApplication.exe”. A splash screen (Figure 4-5) indicates that the application is loading. The prototype already contains the composite application model and necessary data for this walkthrough.

Figure 4-5: Splash Screen of JIGSAW prototype.

4.2.2 Modeling a Business Process using Business Process Modeling Notation (BPMN)

Once the prototype has finished loading, the business process modeling environment is shown in the maximized application window (Figure 4-6). Users can inspect, create, and manipulate the business process model in the Process Canvas. For the purpose of this demonstration, the Process Canvas already contains the model of the customer complaint review process.

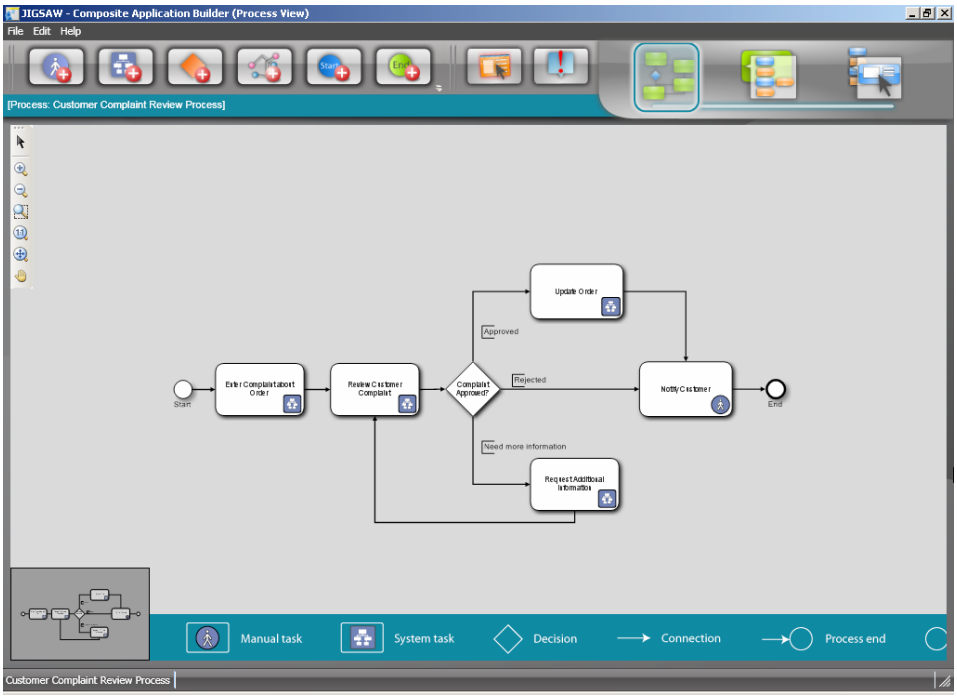


Figure 4-6: JIGSAW's business process modeling environment. The process canvas contains the model of customer complaint review process.

The user wants to add another System Task to the process to support the review of the customer's complaint history prior to the review of the customer complaint. To do so, the user first selects all process elements except for the "Enter Complaint about an Order" task by dragging a rectangle over the process model. The selected elements can be dragged to the right to create a space for inserting the new task. By clicking on [Add new system task] in the toolbar, the user can add the new system task to the canvas by performing a mouse drag gesture in the free space between "Enter Complaint" task and "Review Customer Complaint" task. Once the symbol has

been created on the canvas, the user can double click the element to enter a new label for the symbol (see Figure 4-7).

Figure 4-7: Enter a label for the new system task.

Thereafter, the user can connect the new task with the rest of the model by using the [Add Connection] tool in the toolbar. The user drags a link from the [Review Customer Complaint History" task to the "Review Customer Complaint] task to specify that the "Review Customer Complaint History" task comes beforehand. Then the user reconnects the link between the "Enter Complaint about an Order" task and the "Review Customer Complaint" task by dragging the end of the link from "Review Customer Complaint] to "Review Customer Complaint History]. Figure 4-8 illustrates the result of the reconnection.

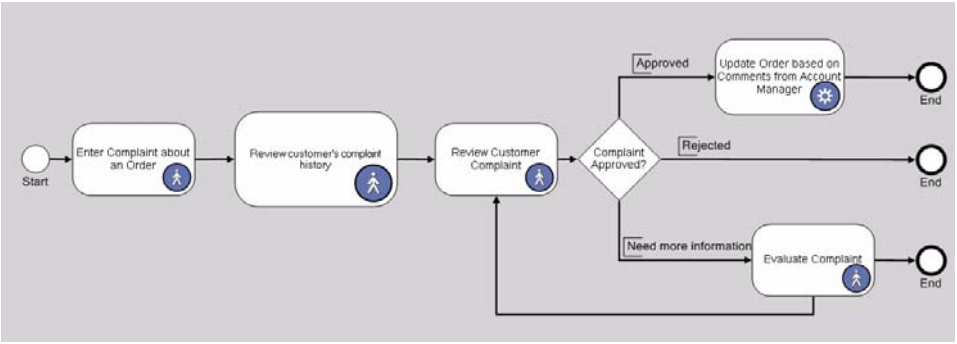


Figure 4-8: The finalized process diagram.

4.2.2.1. Design Aspects

Structure the composite application using Business Process Modeling Notation (BPMN)

To tightly integrate composite application modeling and business process modeling practices, JIGSAW employs the standard business process modeling notation (BPMN) as the primary way to visualize the high-level structure of the composite application. One of most important findings that came out of our user research is the importance of business process flow diagram. The current SAP tools users almost always use other modeling tools such as ARIS and Visio to first create a flow diagram conveying the high-level information of the business process (Flow Model, Appendix A). With familiar notations and environment, business process experts (BPXs) can model in the most effective manner.

During user testing, users also pointed out that a visualization of a business process should look like the standard flowchart they are familiar with (Appendix D).

Additionally, the value of using standard notations is increased as it also allows the model to be easily understood by other stakeholders.

However, given that JIGSAW serves beyond the business process modeling domain, meaningful augmentation of standard notations can be beneficial. Particularly, we introduced an additional icon, namely “Manual Task” and “System Task”, to visually distinguish the tasks in which the workflow is supported by the composite application and offline tasks such as paper work and thought work. This distinction allows users to model manual tasks that do not involve interacting with the system from the system tasks that are backed by a composite workflow, while maintaining the comprehensiveness of the high-level business process model.

In-place editing

JIGSAW allows users to edit the process flow element's name by simply double-clicking on the element. We observed from user studies when testing the lo-fi paper prototype that almost all users perceive double-clicking as the most intuitive interaction to get into editing mode (Appendix D and E).

4.2.3 Editing a composite application workflow

The business process expert wants to extend the workflow of the “Review Customer Complaint” task by adding automatic email notification to review experts to the existing workflow. From the process modeling environment, the user can drill down into the task either by bringing up the context menu by right-clicking and selecting [Edit this task] (Figure 4-9) or by selecting the task and then dragging the navigation slider at the upper-left hand corner from Process Model to Worklfow Model (Figure 4-10).

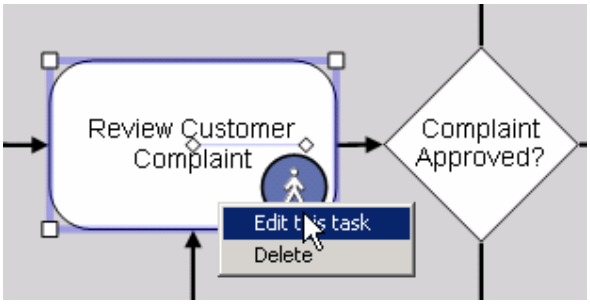


Figure 4-9: Right-clicking a task shows a context menu with functions to edit or delete this task.

Figure 4-10: The navigation slider can be used to quickly switch between the process modeling and the workflow modeling environment.

The window now contains a second diagram canvas above the process canvas (Figure 4-11). This new work area, referred to as the “Workflow Canvas”, is used to edit the

workflow within the selected system task. The lower process canvas, which previously allowed the user to edit the process flow, has now changed to be read-only, while the user can still navigate in the process model.

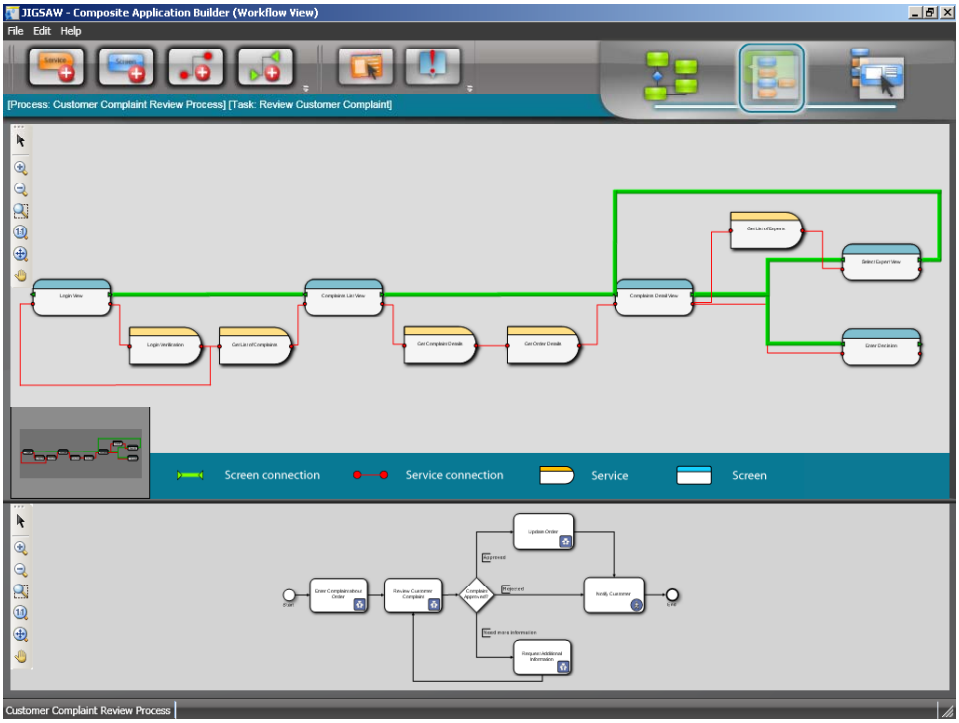


Figure 4-11: The workflow canvas with the read-only process canvas at the bottom of the window.

The Workflow Canvas shows part of the “Review Customer Complaint” workflow. It consists of two different types of symbols: User Interface Screens and Web Services, which are interconnected by green and red links. Green links define the navigational flow between screens and red links define how data is passed from one diagram entity to another.

The user wants to add the email notification component to the end of the workflow. The end of the workflow can be brought into view by panning the canvas to the left. To do so, the user selects the pan tool from the vertical navigation toolbar on the left-hand side of the canvas. The cursor changes to a hand symbol to indicate that the user can perform mouse drag gestures on the canvas to shift the current viewport. The user drags the workflow model to the left to reveal the right-hand end of the model.

The user wants to automatically send an email when an expert needs to be involved in the decision making process. Automatic email sending is a generic service available in the Service Library, which can be opened by clicking on the [Add service] button in the top toolbar of the window. The library opens at the left side of the workflow canvas (see Figure 4-12).

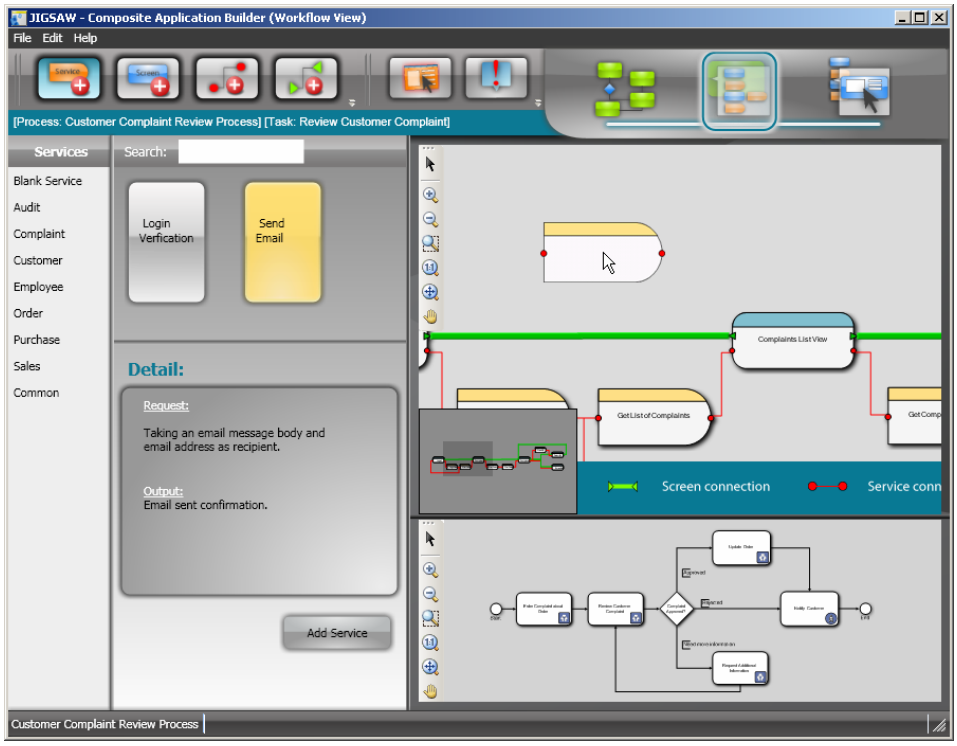


Figure 4-12: The Service Library showing common services

The "Send Email" service can be found either by entering appropriate search terms to the live search box at the top of the library pane or by browsing through the service categories on the left side of the library. The service can be added to the Workflow Canvas either by clicking the [Add Service] button in the detail description box or drag-and-drop the service preview to the canvas.

The user also wants to add a confirmation screen to the workflow to indicate that an email has been sent. To do so, the user opens the UI screen library by clicking the [Add UI Screen] button in the top toolbar then adds a confirmation screen to the workflow from the library in a fashion similar to adding service (Figure 4-13).

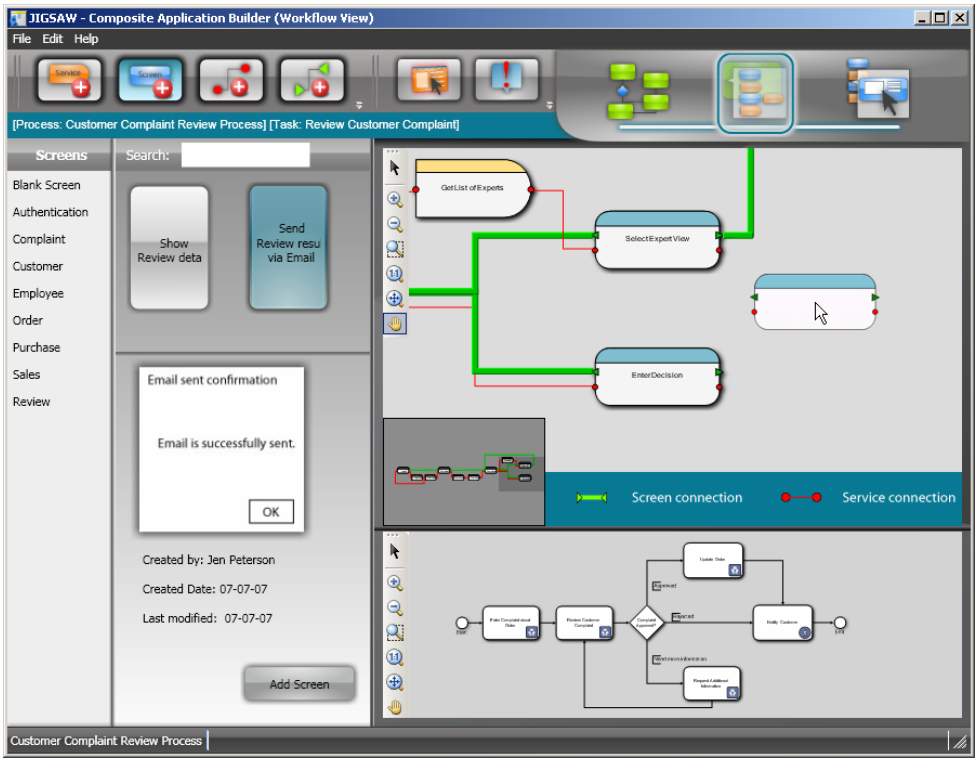


Figure 4-13: The UI Screen Library.

After adding the new service and the new screen to the canvas, the user can use the navigation flow connection tool and data flow connection tool in the top toolbar to incorporate the new workflow elements into the workflow.

The user also wants to ensure the sanity of the workflow that just got changed. The user turns on the sanity check feature by clicking the [Validate the Model] button in the top toolbar. Errors in the model are now highlighted in the canvas (Figure 4-14). The user follows the hint given to fix the errors.

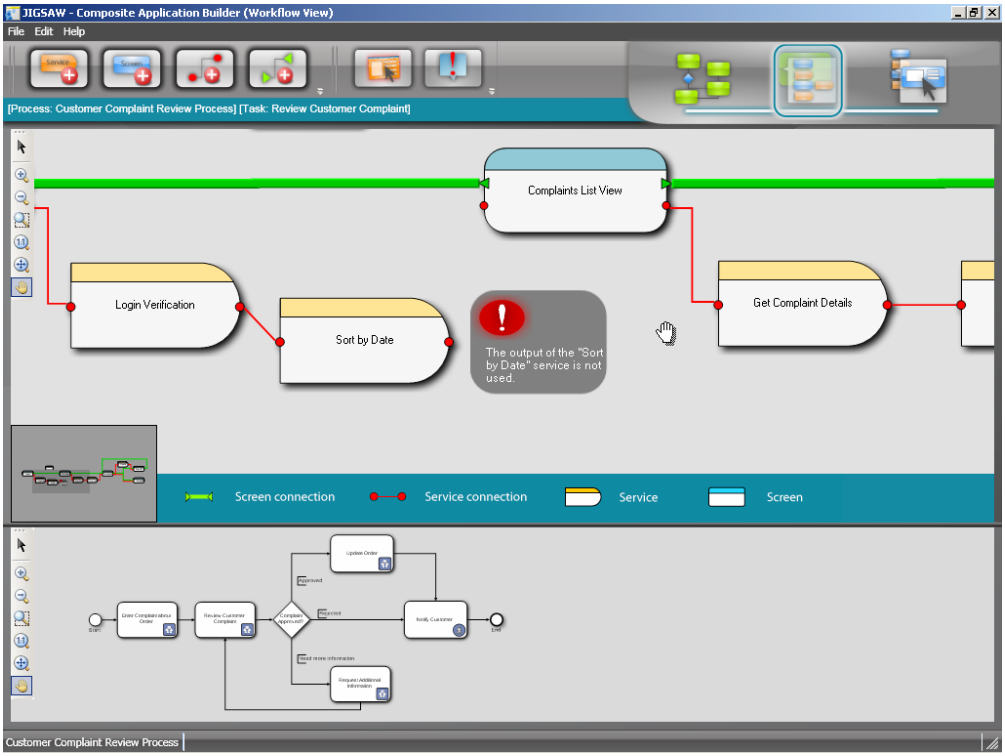


Figure 4-14: The workflow integrity validation detects and highlights the errors in the workflow.

4.2.3.1. Design Aspects

Workflow visualization

We have customized the flowchart symbols and structures to represent a composite workflow. In the workflow model diagram, there exist two types of elements, screens and services. A screen represents a human step in the workflow such as a web form where an end-user would enter customer information; a service represents a system step such as retrieving customer data from the customer database. The relationship between these elements is established by two types of connections: navigation lines and data flow lines. Navigation lines link screens into a sequence of manual actions that an end-user would take to complete the task; data flow lines link output of one service element to the input of another service element.

We make use of contrasting color, shape, and stroke size to visually distinguish different elements and connections. The BPXs stressed the importance of being able to quickly identify and distinguish the human steps from the other components of the composite as these components dictate the end-user behavior of the composite (Appendix E). Data flow visualization makes it easy to depict where the data comes from and where the data goes. Due to the fact that composite applications are built on top of a data-driven platform [11], the data flow forms the main part of the functional specification of a composite. Finally, we provide a legend (Figure 4-15) to assist and remind users of the different symbols due to their unconventional nature.

Figure 4-15: Workflow visualization legend.

We leverage several workflow-modeling paradigms from Visual Composer, in spite of the fact that Visual Composer was originally designed to target business analysts, which coincides with our target users. However, we simplified a number of schemas, such as the concept of layers and the distinction between data binding, data flow, and data mapping. Most of the complexity in terms of configuring screen layout and data mapping is hidden into the lowest level, which is not included in our prototype due to scope and time constraints.

Libraries host examples and foster reuse of best practices

JIGSAW allows users to add elements to the workflow canvas through the screen library and service library. These libraries are designed to host generic examples of screens and services a business process expert would need to specify a composite application. We learned from user research that people never start designing a new process or application or user interface from scratch. They often based their design on existing work that serves a similar function. A number of end-user programming studies also indicate that seeing examples helps users make design decision.

To reduce the amount of technical knowledge required by users in the task of selecting the right web services and application function call, we proposed several features and guidelines in the design of the library:

- The description box presenting details and functionalities about the highlighted action would assist users in selecting the most suitable component. As pointed by our user study participants, it was important to include information such as the last modified date and creator in the description as there is a high possibility of multiple versions being present. Moreover, users prefer to validate the credentials of a creator before they make their decision on which component to reuse.
- The live search would release users from the burden of browsing through the complex tree structure to find the desired components.
- The labels of the taxonomy and actions should be in an easily understandable fashion. We discovered through heuristic evaluation that jargons and technical markup language in SAP's current design-time tool introduce severe usability breakdowns for non-tech savvy users.

We also considered providing pre-built workflows in the library. However, due to scope constraints, we were not able to implement these features in our final prototype.

Our user testing results indicate that the value in live search is well-recognized and a good description and preview of the components is tremendously helpful in selecting the right component (Appendix F).

Automatically validating process/workflow integrity

JIGSAW offers the capability to detect whether there is a flaw in the process or workflow model. A flaw implies workflow mis-connection, dangling tasks, screens or services. Based on our study, once the complexity of the process or workflow increases, it is difficult to spot these problems (see Appendix E for testing results).

We have tested the concept of error detection in our concept validation and user testing. We identify that the validation should happen before moving on to testing mode to reduce the amount of back-and-forth between editing mode and testing mode. It was expected that basic integrity problems, or compile-time errors, be captured before going into runtime.

Navigation toolbar and overview map

One of the biggest drawback of visual programming language is the lack of screen real estate. We attempted to utilize zoomable user interface (ZUI) to provide "virtually unlimited screen space for views" [10]. Our paper prototype user testing results support the acceptance of a zoomable user interface. During these sessions, participants clearly expressed that they want to zoom in on part of the model and zoom out to get the overall picture.

Figure 4-16: Navigation Toolbar provides panning and zooming functionalities related to the canvas.

The navigation toolbar (Figure 4-16) contains functionalities that helps users to navigate and orient the canvas, such as zoom-in, zoom-out, zoom-to-fit, pan, and others. A navigation toolbar is positioned in every canvas to establish its effective use within the specific canvas. The vertical orientation of this toolbar avoids distraction from the main toolbar, as well as fortifies the distinction between the functionalities of the two toolsets.

Figure 4-17: Navigation overview map.

We further implemented the overview map (Figure 4-17) to assist users in navigating around the model. The advantage of overview map lies in the feed-forward of the resulting location in relation to overall model when users use the map to pan. Users have responded extremely positively about the overview map during user testing, and were surprisingly comfortable in using them to pan instead of resorting to scrollbars (Appendix G).

Navigation Slider and Navigation Path

To quickly switch between the process modeling, the workflow modeling, and the testing environments, users may use the Navigation Slider (Figure 4-10) on the top right corner. The slider reveals the metaphor of increasing level of detail when going from the process level to the workflow level and then to the screen and service testing level. We also made the state of the slider large and prominent to remind users which environment they are currently in.

Figure 4-18: Navigation Path showing the context of the current environment.

Similarly, the Navigation Path (Figure 4-18) right below the toolbar provides single-click tracing back to the levels above. Additionally, it provides more description of the hierarchical context by naming the process or task the user is currently in. This feature is consistent with the one in Visual Composer. It was also requested and validated by the users during user testing of the paper prototype (see Section 6.6 for detail).

Read-only process canvas

Keeping the process canvas read-only prevents distracting users from workflow editing. By imposing task separation (i.e. process flow editing and workflow editing), users benefit from a simpler development environment. More specifically, by having the contextual toolbar display only the functionalities that are relevant to the workflow-editing task, we lessen the user's mental workload by pre-filtering out the inapplicable options.

4.2.4 Testing the workflow

Modeling a workflow from user interface screens and services results in a new composite application. The changed “Customer Complaint Review” workflow can instantly be tested within the JIGSAW application.

The user switches to the testing environment by dragging the navigation slider to the right from [Workflow Modeling] to [Testing]. The testing environment shifts the workflow canvas to the bottom of the screen to make space for detailed views on screens and service instances (see Figure 4-19).

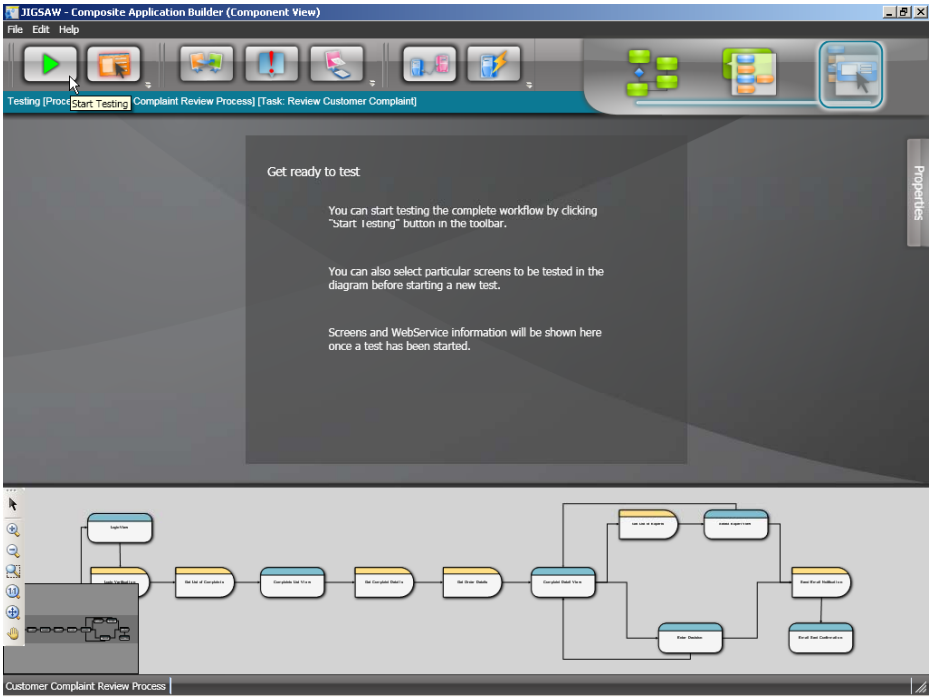


Figure 4-19: The workflow test environment before starting a test.

The user can select entities in the workflow model to see details of both user interfaces and services. Services are represented by showing input and output parameters plus details about the service implementation (see Figure 4-20).

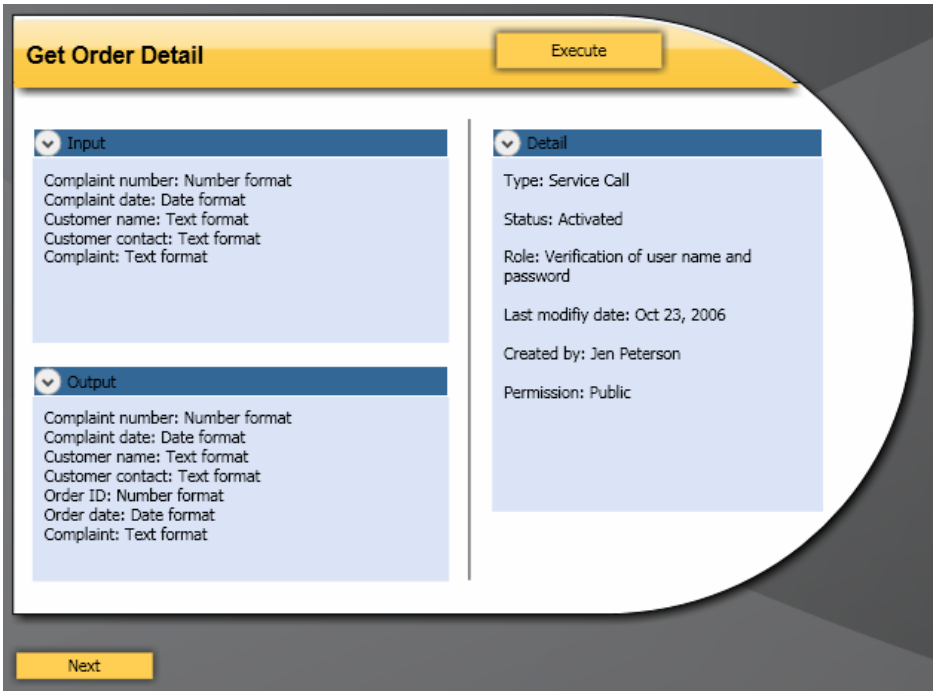


Figure 4-20: Service detail view in the workflow test environment.

The user starts a test by clicking the [Start Test] button in the top toolbar of the window. Before starting the test, however, users can use the Property Sheet (Figure 4-21) to change the environment variables such as roles, permission, and time. To access the Property Sheet, the user clicks the [Properties] tab at the right-hand side of the canvas.

Figure 4-21: The Property Sheet in Test View

When the user clicks the [Start Test] button, the [Activate Service Call] dialog box shows up (Figure 4-22). The user can select the service calls that will be executed and deselect the services that will only be simulated during this test session.

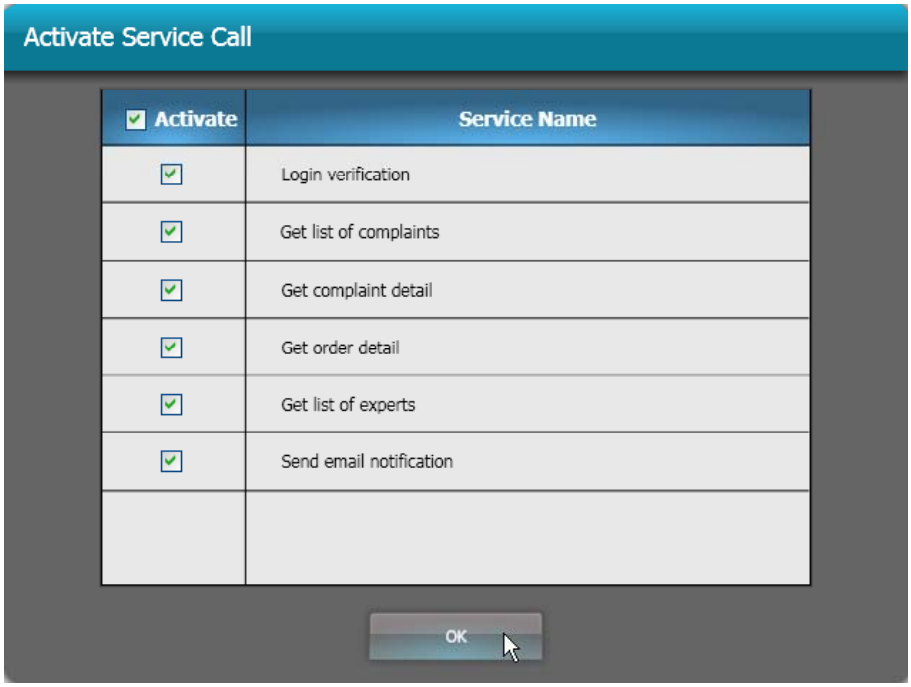


Figure 4-22: The service activation dialog box.

Upon the start of the testing session, the content of the first step of the workflow is displayed and the step is highlighted in the Workflow Canvas. The user interacts with the displayed screens or executes the services to test out the composite application (see Figure 4-23 for an example).

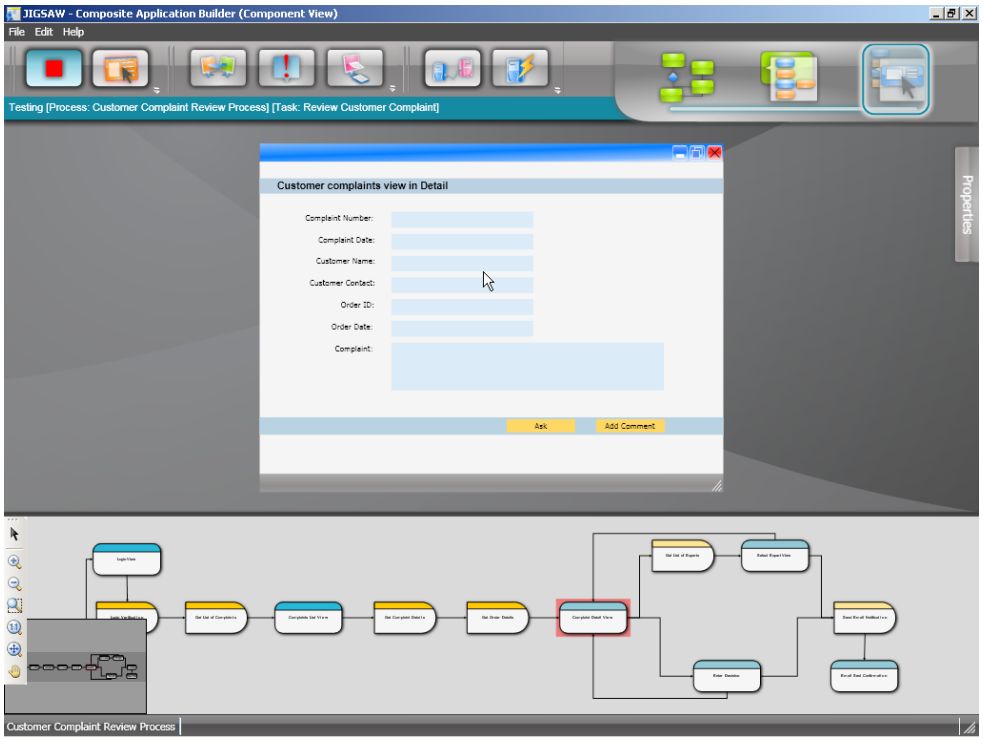


Figure 4-23: An example of a screen in the testing environment.

When the user has questions about where the data comes from and where it goes, the user can turn on the service dependency feature (see Figure 4-24).

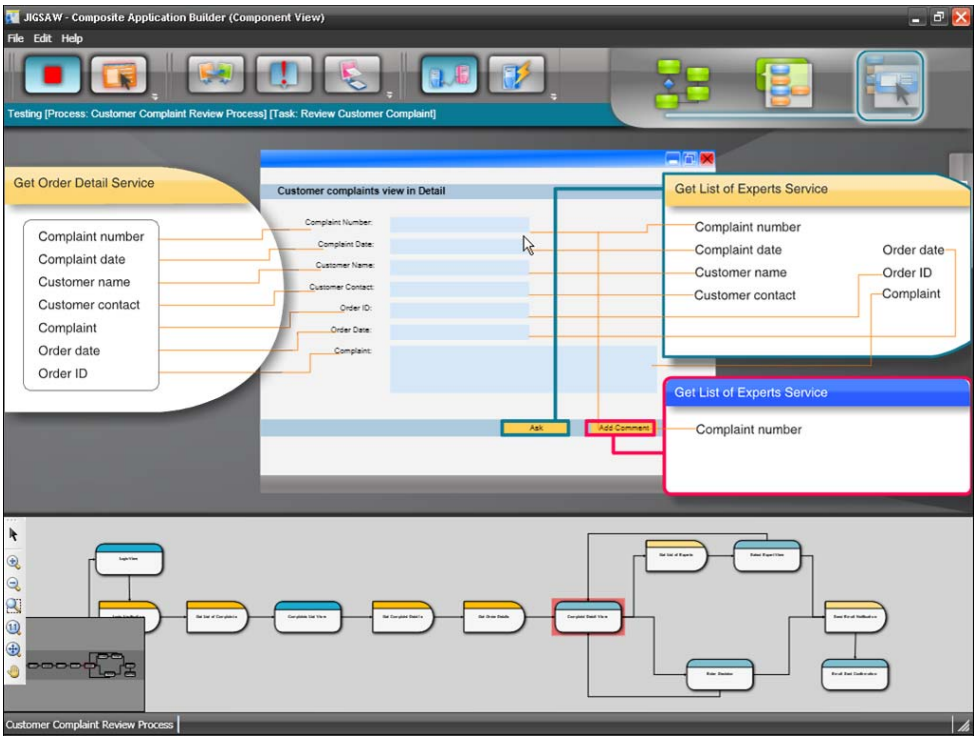


Figure 4-24: Service Dependency View

Once a path through the workflow has been tested to the final step, a test report is shown to the user (see Figure 4-25). It summarizes the test case by listing all screens and services visited during the test. It reports the time screens were displayed and the time needed to complete service requests. It also reports input and output parameters of both screens and services.

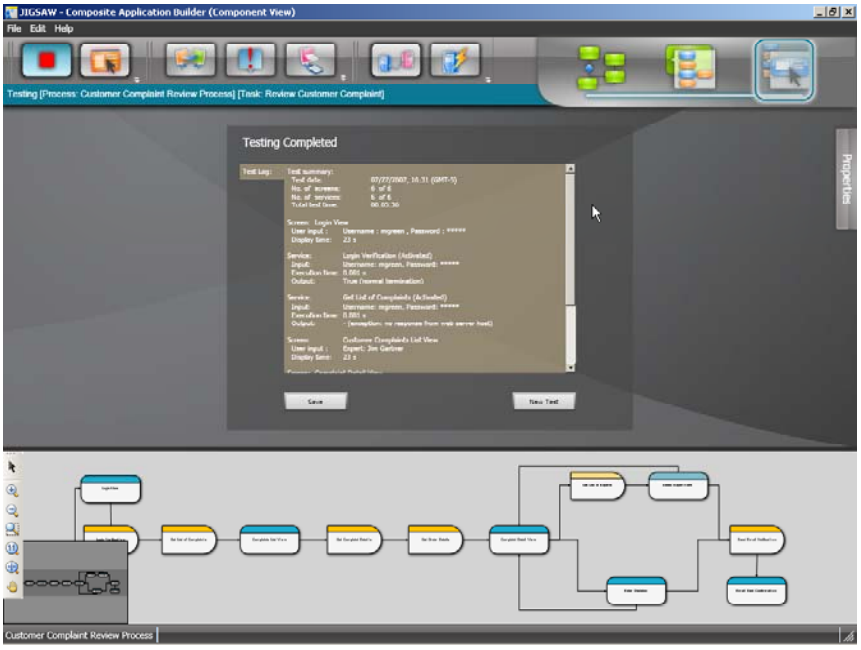


Figure 4-25: Summary report of a test case.

The user can save the report and start another test session or return to the modeling environment to make changes to the model. When the user starts another test, the test environment will hold the test parameters previously entered.

4.2.4.1. Design Aspects

Selecting to preview specific step in Workflow Canvas

In addition to providing contextual information, users can also use the read-only Workflow Canvas to select certain steps in the workflow to preview the screen or retrieve service information before starting a test session. This feature allows user to freely inspect the details of each step, which is a feature requested by our user testing participants (see Appendix G for details). After the test starts, users are no longer allowed to jump ahead in the workflow because it is expected that users would like to simulate the experience of the composite application to be as real as possible.

Setting environment variable using Property Sheet

We learnt from our user studies that it is essential to be able to change the testing environment variables such as the user roles and permission and system date and time, because the services may react differently based on these variables (see Appendix E for detailed user feedback). Nevertheless, users would not like to be bothered by the setting every single time they start a new testing due to the highly iterative nature of process design and testing. Therefore, we keep the default settings to be the same as last test and have them in the Property Sheet minimized to the side where it is easily accessible.

Inspecting service dependency

JIGSAW enhances the visibility of data flow by allowing users to view the detailed data mapping between the input and output data structures of services and the dynamic fields in the screens. The visualization helps users to gain a better understanding of how data is populated and consumed and to debug any unanticipated composite behavior due to auto-mapping during workflow modeling.

Services activation and simulation

We learnt from user feedback that there are a number of good reasons why a user would not want to actually execute the service call during preliminary composite application testing (see Appendix E and F for details):

1. The transaction may cause unwanted changes to the back-end system
2. The service may be inactivated or not available due to various reasons
3. The service call may take too long to execute

Hence, JIGSAW provides users with an option to select which service to activate to preempt function call execution. Although not captured in our final prototype due to scope limitation, we meant to allow users to enter simulated data for the deactivated services calls. We can also imagine that such a feature can further empower business process experts in modeling dummy services and further specifying the dummy services behavior through defining the input and output.

For the active services, we expect users to manually start the service execution during testing when it is the time to use the service. This serves as an extra layer of screening in preventing irreversible transactional error. It also helps users to better understand the service behavior.

Publishing prototype for user acceptance test

During our user research, we discovered that a new business process would always be tested by multiple stakeholders before being put on the production system (see Appendix A for details). JIGSAW supports this practice by offering the ability to publish and share the resulting prototype of composite application model to other stakeholders for preliminary user acceptance test. The prototype may also be used as an artifact in communicating and promoting the new business process to the end-users.

Test report provides basis of evaluation and improvement

Our user research revealed that business process experts are mostly interested in statistical data that proves the effectiveness of the new process (see Appendix C). After every testing session, JIGSAW aggregates the parameter data (i.e. input and output) and statistical data (i.e. time spent at each step) into a test report. Business process experts can then use this data to analyze process efficiency. This is particularly helpful when the experts publish the prototype to end-users. JIGSAW also allows users to export these workflows into other formats (e.g. PDF, Microsoft Word, plain text) that can be shared and documented easily.

5 Interactive Prototype

5.1 Development Environment

5.1.1 Choosing a development environment

Before choosing a development environment for the JIGSAW prototype, the group thoroughly evaluated the factors and constraints which influence the choice of a prototyping technology.

Expertise of team members

We evaluated the expertise of the team and found that the team has expertise in prototyping based on both web technologies and rich-client technologies. All team members had experience developing web pages based on CSS and HTML and some members had experience using JavaScript as well. A majority had also expertise using Flash, but had only novice knowledge of ActionScript programming. Most team members also turned out to have a working knowledge of programming with VisualBasic.NET and C# due to mandatory courses in our program of study.

Functional Requirements

We identified potential functional requirements early in the design process.

- User research pointed out that business process experts often use flow diagram notations when creating and communicating business process designs. Early on, we identified strong value in this work practice, and developed multiple different concepts involving end-user programming environments based on *flow-charting* techniques which require highly interactive drag and drop functionality.
- We have also found in user research, that business process experts have varying needs when visualizing and communicating business process designs and ERP applications. Some have to evangelize end-users for them to buy into a particular process design or software system. Others strategically streamline and optimize business processes on a rather high level of

abstraction to increase a company's long term revenue. We therefore considered a *zooming user interface* a vital feature for a composite application design environment which allows a user to quickly switch between multiple levels of abstraction.

Both functional requirements; an interactive flow-charting design environment and a zooming user interface, would have been difficult to implement in a pure web environment. Such requirements would be hard to meet even by applying the latest AJAX techniques which promise rich user experience in web applications.

Integration of design and development processes

For the prototyping purpose, we also determined the ability of a technology to gracefully integrate with hi-fi design artifacts as essential to achieve fast implementation.

Client expectations

Our client did not impose a technology for prototype implementation, but expressed the wish to receive an interactive prototype as major design deliverable. The client used the metaphor of a "concept car" to describe the ability of a prototype to communicate a user experience in a rich way.

5.1.2 Microsoft Expression Tools

The team decided that the Microsoft .NET technology best fits with the constraints and factors outlined above. The development environment for the .NET Framework 3.0 provides solid design tools [3] for the implementation of highly interactive software systems and gives almost unlimited flexibility when designing a custom look and feel. It provides built-in support for zooming user interfaces and includes a solid drag and drop framework. The team also brought the fundamentals needed to rapidly learn the new tools and technology.

5.1.3 ILOG Diagrammer Program Library

Results from user research inspired design ideas and concepts based on flow charting techniques to define business processes and workflows. The team identified such prototype functionality as a high risk for the implementation phase as it requires extensive programming effort when implemented from scratch. We therefore researched ways to provide flow charting functionality in the prototype similar to Microsoft Visio without the extensive programming overhead.

In our research for a re-usable component for the .NET Framework, we came across ILOG Diagrammer 1.0 for .NET. When used in a .NET WinForms application, this program library provides GUI components to interactively display, create and modify diagrams of various types. We decided to use this program library for the JIGSAW prototype after developing a conceptual prototype which highlighted important functionality such as direct-manipulation of diagram elements with drag and drop, navigation in a diagram through zoom and pan operations and support of custom diagram symbols.

5.1.4 Experience with development tools

The tools we used enabled the whole team participate in prototype development. The team effectively divided work between designers and programmers.

Designers created high-fidelity designs using Expression Design and Expression Blend tools. These tools allow designers to create designs including user interface controls

and vector graphics while producing user interface descriptions as a XAML document. XAML, short for eXtensible Application Markup Language, is a declarative XML-based language [12] used to define user interface components without writing any line of traditional source code.

Designers handed user interface components to developers who then added application logic to bring the user interfaces to life. They used Microsoft's Visual Studio 2005 to wire the user interface components with functionality to achieve a highly interactive prototype.

This development environment does also encourage Pair Programming as known from the extreme programming methodology [8]. Both Expression tools and Visual Studio can be run in parallel to let both designers and programmers sit together and develop software artifacts in pairs. Team members found this type of collaboration to be a very encouraging experience; the work done in pairs is of high quality while designers and developers rapidly learn from each other.

5.1.5 Experience with ILOG Diagrammer

Using ILOG Diagrammer was very beneficial as it allowed us to rapidly add sophisticated flow charting functionality to the prototype without programming overhead.

In order to be effective in rapid application development, a 3rd party program library needs to be well documented, tightly integrate with development tools and should also ship with comprehensive sample code. All these features are available with the ILOG Diagrammer product. It includes a detailed API reference and tutorials, integrates very well into the visual user interface builder within Visual Studio 2005 and also ships with solid sample code.

However, the team also faced some challenges when using ILOG Diagrammer for the JIGSAW prototype implementation. The ILOG program library is developed for the .NET Framework 2.0 which did not match our development environment of choice. The Microsoft Expression tools are used to develop for the Windows Presentation Framework (WPF) of the .NET Framework 3.0. We therefore used a generic technique provided by Microsoft [14] to host the ILOG components within the JIGSAW prototype. The integration of the ILOG components through this hosting technique was not problematic from a coding perspective. However, we realized during development that the ILOG components were always drawing on top of the layout structure, overriding the drawing of WPF components. Moreover, the update of the layout structure with mixed user interface components seemed to cause slight flickering effects. The flickering effect can be observed when switching between application modes using JIGSAW's navigation slider in the top right of the application window. We found that the visual layout problems described are not caused by the ILOG user interface components, but are side-effects caused by mixing user interface components based on the .NET Framework 2.0 within a .NET Framework 3.0 application. Further investigation of the layout update mechanisms within the Window Presentation Framework might help to resolve this issue.

5.1.6 Limitations

5.1.6.1. Limited ILOG License

The diagram design capability of the JIGSAW prototype was implemented using a commercial program library from ILOG [5]. Free licenses have been provided by ILOG for the duration of this project. These development licenses are no longer valid after 31st of August 2007.

5.1.6.2. Adding Animation

The Microsoft .NET Framework 3.0 provides a framework to create animations within rich-client applications similar to Adobe's Flash. However, a designer used to Flash finds the capabilities of the .NET Framework to be rather limited to date. The Microsoft Expression Blend tool does provide a timeline view of animations similar to Flash, but makes it hard for the user to visually define an animation through direct manipulation. Instead, single properties of user interface components are animated, which reduces the animation to a bunch of changing variables.

5.1.6.3. Designing custom button styles in Blend

Buttons with a pleasing look and feel seemed to be easy to create on a first glance, but turned out to be complicated controls when all requirements were taken into account. Buttons should at least provide visual feedback when being pressed, but buttons should also visually indicate when they are focused/unfocused and disabled/enabled. These states guarantee that the user quickly recognizes whether the button can be used and whether it accepts a keyboard command such as "Enter". The affordance of a button control can also be improved by changing its appearance on mouse over.

Changing visual states of a button can be implemented using the Blend's animation capabilities. Triggers have to be defined for a button to fire an animation upon state change. The associated animations define how the appearance of a button changes from one state to another. Each custom button style therefore requires a designer to define both triggers and animations to achieve the desired visual feedback when the button's states change.

The difficulty comes in when these states overlap: buttons can still be pressed when the mouse does not hover over it (at least on Windows operating systems) and buttons can also lose their focus while being pressed. Designing a custom button style in Blend is complex and therefore requires attention to detail and should explicitly be verified through user tests.

5.1.6.4. Drag and Drop interaction between WPF and ILOG components

We implemented Drag and Drop interaction for adding services and screens from the library onto the workflow canvas. However, this solution does not adhere to the common drag and drop interaction scheme. Instead, we allow a user to pick a service or screen from the library and then change the interactor of the workflow canvas to show a ghost under the cursor. When the user clicks, the corresponding symbol is added to the canvas at the location of the cursor. The common drag and drop interaction scheme, where an object is dragged and finally dropped over a drop target could not be implemented between WPF and ILOG components. This functionality is confirmed to be buggy according to an official MSDN Forum thread [15].

5.2 Prototyping Technology

5.2.1 Installing the prototype

5.2.1.1. Overview

This section explains in detail how to run the JIGSAW prototype. Both source code and binaries of the prototype are made available to our client as an internet download. However, neither source code nor binaries will be available through the official project website.

5.2.1.2. Operating System

The prototype has been developed and tested on the Windows XP SP2 operating system, but should also be executable on a Windows Vista operating system. Either one of them is required to successfully run the JIGSAW prototype.

5.2.1.3. Installing .NET Framework 3.0

The prototype is based on the Microsoft .NET technology and requires the .NET Framework 3.0. The .NET Framework 3.0 is readily shipped and deployed with the Windows Vista operating system. The .NET Framework 3.0 needs to be separately installed on a Windows XP SP2 operating system.

Please refer to the following webpage to download the .NET Framework 3.0 installer (ca. 50MB) for x86 processors:

http://go.microsoft.com/fwlink/?LinkId=70848

Double click the file to start the installation when the download successfully completed.

If you do not know whether you have .NET Framework 3.0 installed on your system or not, go to the following path and see if the folder "v3.0" exists:

`%systemroot%\Microsoft.NET\Framework`

If so, you have .NET Framework 3.0 already available.

5.2.1.4. Installing ILOG with a SITE license

The JIGSAW prototype requires the ILOG Diagrammer 1.0 for .NET software component with a valid license.

ILOG Diagrammer 1.0 for .NET is provided to our client as a separate download package. To run the installer, execute setup.exe from the downloaded package. The installer asks to enter license information. You find the two lines of text required to specify this license in the access.ilm file also located in the same download package.

The development SITE license provided in the download package is only intended to be used by our client for the purpose and duration of our project. This license has been provided for free by ILOG and is valid until August 31st, 2007.

A valid license for ILOG Diagrammer is required to run the JIGSAW prototype. After August 31st, 2007, a valid license for ILOG Diagrammer can be purchased or an evaluation license key might be obtained from ILOG at http://www.ilog.com.

5.2.1.5. Installing the prototype

The JIGSAW prototype is provided as a ZIP file ("JIGSAW-Prototype.zip") for download. This ZIP file contains a readme file which explains the contained source code structure and compilation with Visual Studio .NET 2005. However, the prototype does not require compilation or an installation procedure.

The file "\JIGSAWApplication.v5.4\bin\Debug\JIGSAWApplication.exe" can be run from within the unpackaged ZIP file. The contents of this ZIP file can also be copied to any folder on the local machine before execution.

5.3 Prototype Specification

5.3.1 Overview

The design ideas and concepts found in user studies and the design process have led to an end-to-end solution in the form of a rich-client application, referred to as the JIGSAW prototype. It provides the user with the functionality to design business processes and underlying composite application workflows. Moreover, it provides an environment to immediately test workflows at design-time.

The JIGSAW prototype's user interface (see Figure 5-1) is consistently structured to provide a unified user experience in the different application modes. Figure 5-2 shows how the JIGSAW window is structured into four major sections.

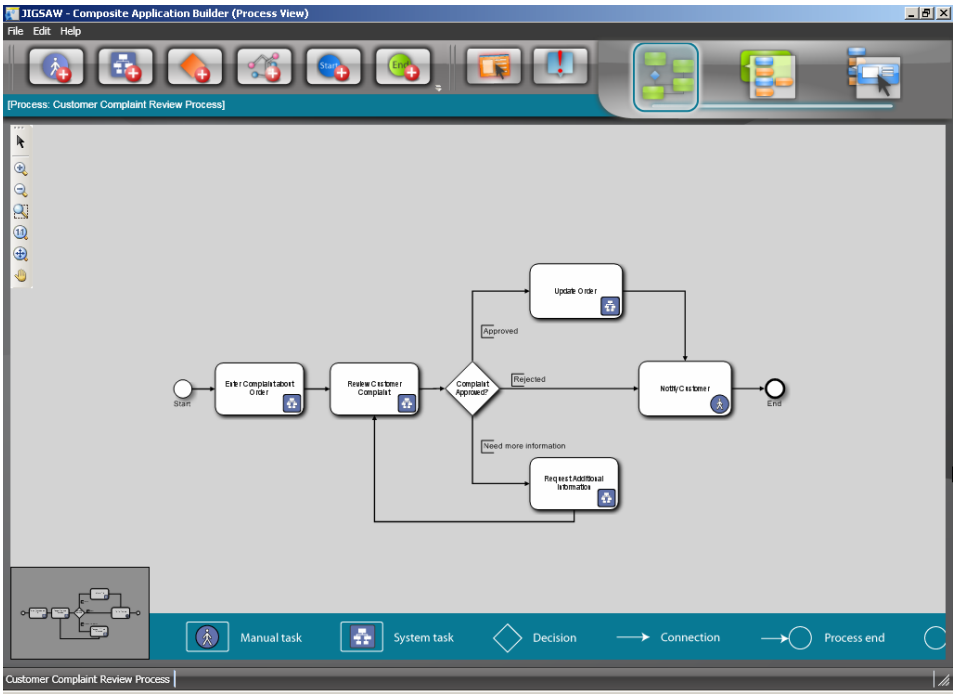


Figure 5-1: Screenshot of the JIGSAW window

5.3.1.1. Menu Bar (1)

The menu bar provides standard functions such as [File]->[Exit] which are available at any time during application run-time.

5.3.1.2. Navigation Panel (2)

The navigation panel provides functionality depending on the current application mode while maintaining a consistent look & feel across modes. The navigation panel includes a contextual toolbar on the left while providing a slider control to switch between application modes on the right. The toolbar always provides functionality relevant to the work area whereas the slider control to switch between application modes remains the same. The navigation panel also includes a breadcrumb bar at the bottom. It reveals what is displayed and its level of detail.

5.3.1.3. Work Area (3)

The work area provides maximized real-estate for the user's current task. The user's task depends on the current application mode. We distinguish three application modes (see Figure 5-3):

- **Process Design Mode**

  The user is designing a business process model which is comprised of tasks which are connected to each other. The work area contains a large process canvas used to spatially arrange the elements of the business process model.

- **Workflow Design Mode**

  The user is designing a workflow for a system task of the business process model. The work area contains a canvas to spatially arrange workflow elements in the upper half and shows the corresponding business process model in the lower half to maintain the user's context of work. A library including workflow elements is available on the left side of the work area when needed.

- **Workflow Test Mode**

  The user is testing a composite application which has previously been designed. The work area is again split into two halves: the upper half is used to visualize screens and service overviews of the workflow, while the workflow model is presented in the lower half of the work area.

5.3.1.4. Status Bar (4)

The status bar indicates which process is currently being loaded in the JIGSAW application. The status bar does not yet provide additional information or contextual help (see section 5.3).

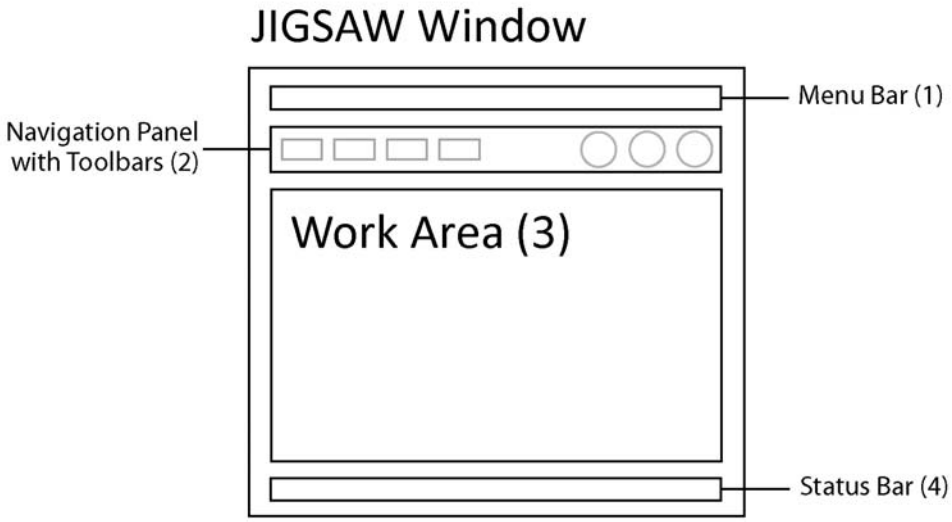


Figure 5-2: Static User Interface Structure of the JIGSAW Window

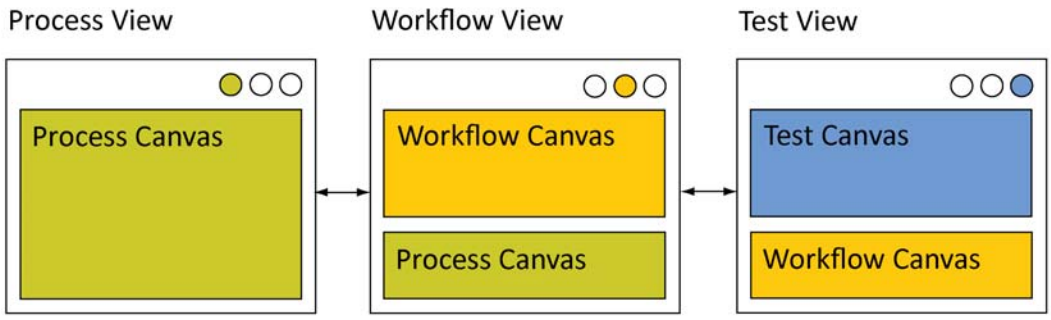


Figure 5-3: The three application modes of the JIGSAW prototype

5.3.1.5. Design Canvas

Both the process canvas and the workflow canvas are used to construct flow diagrams based on different flow diagram notations. They recur in different application modes and share common functionality and common behavior.

Navigation Toolbar

Each design canvas contains a vertical toolbar (see Figure 5-4) in the upper left corner used to navigate in the displayed flow diagrams. Its functions are provided by the ILOG Diagrammer program library and mimic the standard behavior of flow charting tools such as Microsoft Visio. The toolbar functions are described in Figure 5-5.

Figure 5-4: Canvas Navigation Toolbar

Icon	Function Name	Interaction
	Select	Allows user to select one or more elements on the design canvas. Multi-selection requires mouse drag gesture while holding down the left mouse button.
	Zoom In	Zooms into the currently displayed diagram.
	Zoom Out	Zooms out of the currently displayed diagram.
	Zoom into Area	Zoom into specified rectangular area. Requires mouse drag gesture while holding down the left mouse button.

	Original Zoom	Restore original zoom.
	Zoom to Fit	Zoom to fit the diagram into the view port.
	Pan	Drag the diagram while holding down the left mouse button.

Figure 5-5: Navigation Toolbar Functions

Overview

The design canvas which is of primary focus contains a small flow diagram overview (see Figure 5-6) in the lower left corner. It helps the user to recognize the extent of the currently displayed model relative to the displayed portion of the diagram. The user can drag the display bounds within the overview to change the current view port. This helps the user to maintain spatial orientation at any level of zoom.

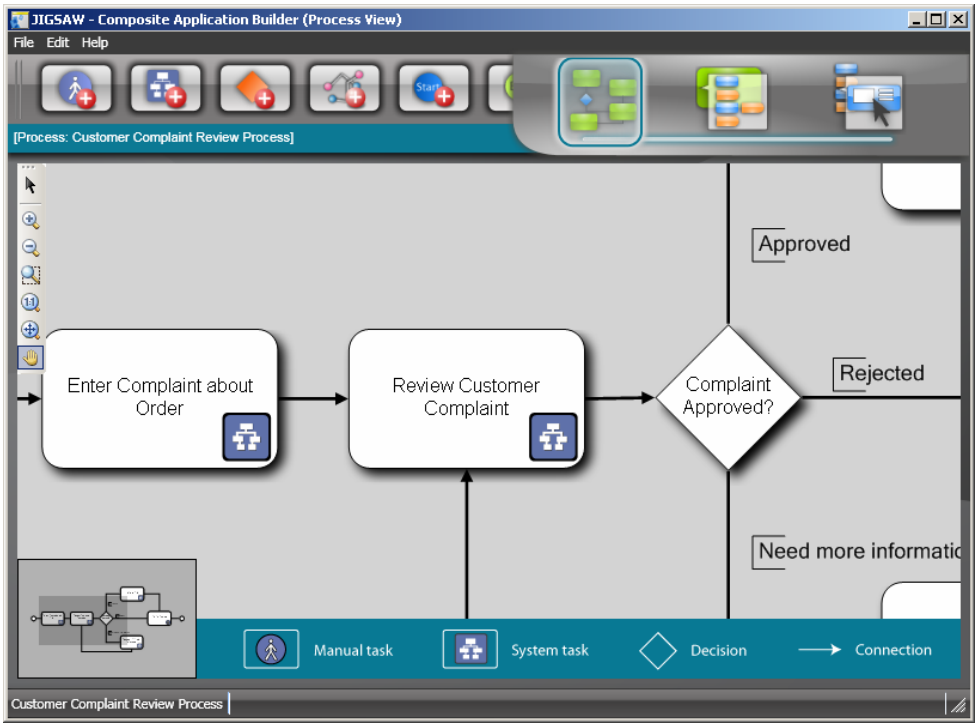


Figure 5-6: The zoomed area is highlighted in the overview panel on the lower left corner of the window.

Legend

The design canvas which is of primary focus contains a legend at the bottom. It gives the user an overview of the elements available in the current design canvas and should help a user to construct a particular type of flow diagram (either a workflow or process model, see Figure 5-7 and Figure 5-8).

Figure 5-7: Legend for Process Canvas

Figure 5-8: Legend for Workflow Canvas

Direct manipulation of elements in the design canvas

If no particular tool is activated in the toolbars, the user can select and move diagram elements. The user can select single or multiple process elements and also move or delete selected elements. Selection using Mouse and Keyboard works as known from Microsoft Visio. However, keyboard interaction has only been implemented for multi-selection and deletion of selected elements. Arrow keys cannot be used to move selected elements on the canvas. This functionality should be made available in a productive system (see section 5.4.2). The labels of diagram elements can be changed upon double click.

5.3.2 Process View

5.3.2.1. Overview

The Process View (see Figure 5-9) is the application mode initially shown when the JIGSAW prototype is started. It allows a user to design business process models based on basic elements defined in the standardized Business Process Modeling Notation (BPMN) [2]. The work area contains a large process canvas dedicated to the process design use case as real-world process models tend to be very large. While testing the JIGSAW prototype with users, we learned that they distinguish manual tasks which represent offline work from system tasks which are supported by software systems. We therefore let users create both manual and system tasks. Tasks can be connected to form task sequence including decisions. Users can also add start and end points to a task sequence.

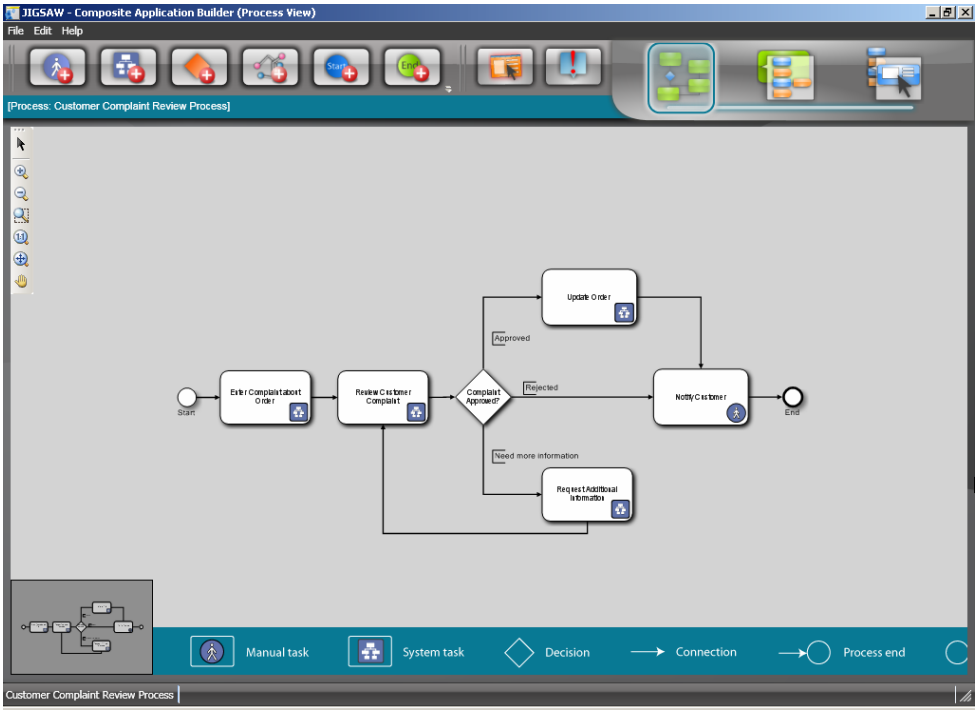


Figure 5-9: The Process View

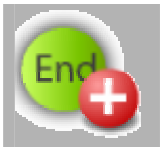
5.3.2.2. Components of the Process View

Several tools and functions are available to navigate on the canvas and to manipulate the process model. In the lower-left corner, a process overview panel (see section 5.3.1.5) lets users browse the canvas in the case that the model can not be displayed entirely in the current view port of the canvas. A vertical and horizontal toolbar are provided in the top-left corner of the window. The vertical toolbar contains controls to navigate on the process canvas (see section 5.3.1.5). The horizontal toolbar

contains functions to create elements of the process model. A legend (see section 5.3.1.5) in the bottom of the process canvas summarizes the elements used to draw business process models.

5.3.2.3. Main Toolbar

The toolbar in the top of the window includes functions to create manual tasks, system tasks, decisions, links, start points and end points. When the user clicks on a toolbar button, the underlying function is activated so that it can be carried out on the process canvas. The function is automatically deactivated when the operation is completed. There is also an [Edit] button and a [Verify] button on the toolbar without attached functionality. Table 5-1 summarizes the functions available in the Main Toolbar.

Icon	Function Name	Interaction
	Add Manual Task	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, the symbol is created in place.
	Add System Task	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, the symbol is created in place.
	Add Decision	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, the symbol is created in place.
	Add Start Point	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, the symbol is created in place.
	Add End Point	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, the symbol is created in place.
	Add Connection	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, the connection is created in place. When dragging a connection end over a shape, connection points become highlighted. When the mouse button is released while a connection point is highlighted, the connection snaps to the

		connection point.
	Edit System Task	Not yet implemented. The user should be able to select a system task and go to its workflow definition by pressing the [Edit] button. For now, this functionality is only provided via context menu entry when right-clicking a system task in the process canvas.
	Verify Process Model	Not yet implemented. By pressing the [Verify] button, the system should check business process diagram rules and indicate rule violations directly on the process canvas in the context of the process model. This functionality is demonstrated in the workflow canvas implementation.

Table 5-1: Functions for creating and manipulating a process model

5.3.3 Workflow View

5.3.3.1. Overview

The Workflow View (see Figure 5-10) provides the functionality to construct workflow models for tasks in the process model. The user reaches this application mode when choosing to edit a task within the Process View by right-clicking a selected system task and choosing [Edit] from the context menu. The work area becomes horizontally divided to show both the workflow canvas and the process canvas. The process canvas remains visible with the appropriate task highlighted in order for the user to recognize that the currently displayed workflow is part of a particular system task of the business process.

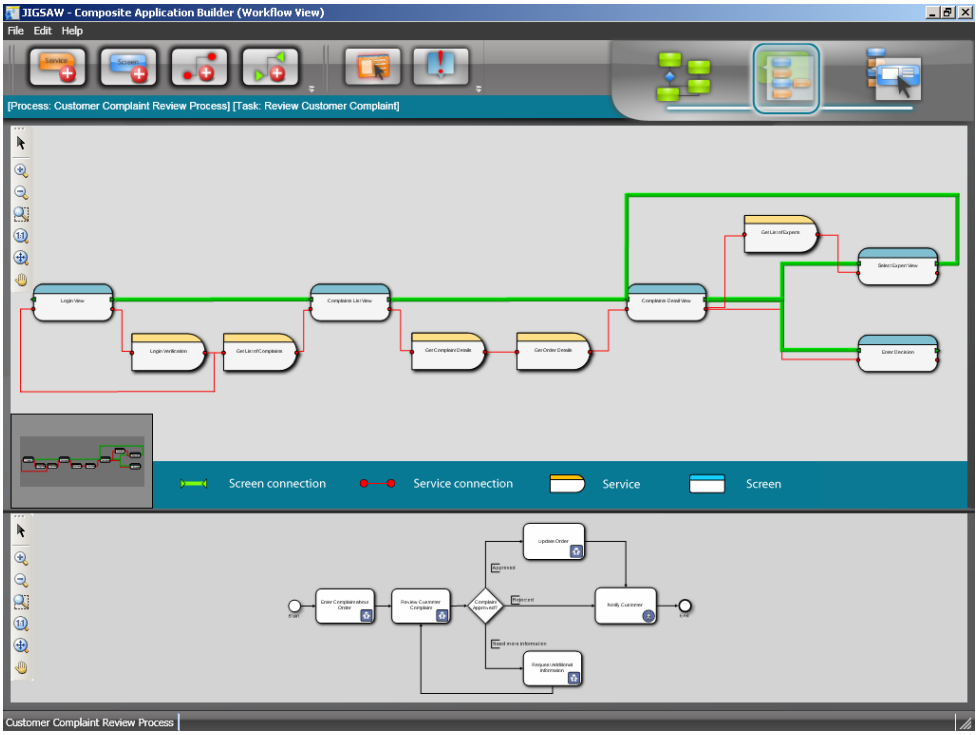


Figure 5-10: The Workflow View

5.3.3.2. Components of the Workflow View

The workflow canvas in the upper part of the work area contains again a diagram overview, a toolbar for diagram navigation and a legend summarizing the available diagram elements (see section 5.3.1.5). The process canvas from the Process View is still visible in this application mode. However, it is read-only and can not be manipulated. The main toolbar primarily provides functions to open screen and service libraries, to create navigation and data flow connections (see Table 5-2 for a detailed listing of available functions).

5.3.3.3. Workflow Canvas

The user constructs a workflow by picking existing services and user interface (UI) screens from a library which is attached to the left side of the work area. Services and UI screens do not adhere to the BPMN notation. Two different shapes and colors are used to distinguish screens from services as shown in Figure 5-11. The blue headed shapes represent screens and contain green and red input and output ports. The yellow headed shapes represent services and contain red input and output ports. The different colored ports are used to define two different types of links between UI screens and services. Thick green links are used to define navigational flow between screens while thin red links are used to define data flow between screens and services. They have been introduced to support the different approaches users take when defining workflows. Some users think of screen flows while others are primarily interested in data flow. By using both different colors and different shapes for workflow entities and port types, we make sure that the user can easily distinguish the different concepts.

A workflow constitutes a guided procedure in the NetWeaver terminology.

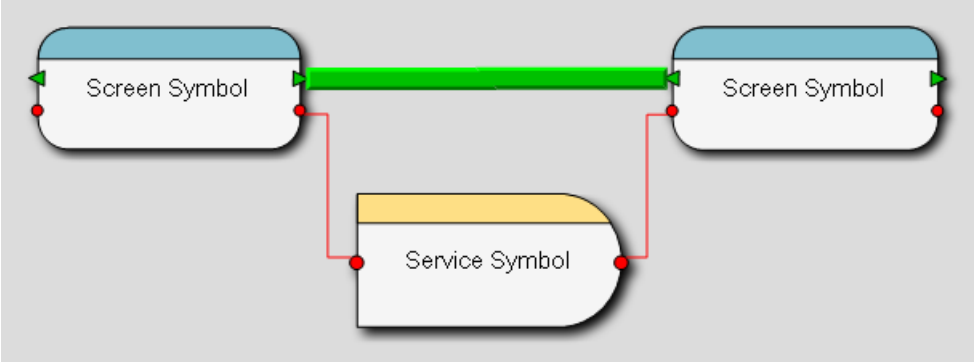
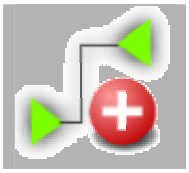


Figure 5-11: Service and Screen Symbols with connections as shown on the workflow canvas.

5.3.3.4. Main Toolbar

The top toolbar (see Figure 5-10) contains functions to add services, UI screens, navigation connections and data flow connections to the workflow canvas. After clicking any of the first four buttons, the corresponding function is activated and ready to be performed on the workflow canvas. Once completed according to the interaction specified in Table 5-2, the function is automatically deactivated to let the user manipulate the new diagram element.

Icon	Function Name	Interaction
	Show/Hide Service Library	When clicked, the service library is shown attached to the left side of the work area. Clicking the button again hides the library.
	Show/Hide Screen Library	When clicked, the UI screen library is shown attached to the left side of the work area. Clicking the button again hides the library.
	Add Navigation Connection	Once clicked in the toolbar, a mouse drag operation on the canvas while holding down left mouse button will create a connection. When releasing the left mouse button, a connection is created in place. When dragging a connection end over a shape, connection points become highlighted to indicate the drop target. When the mouse button is released while a connection point is highlighted, the connection snaps to the connection point.
	Add Data Flow Connection	
	Edit	Not yet implemented. The user should be able to select a service or screen and go to its component view by pressing the [Edit] button. This

		function has not been made available in the JIGSAW prototype. However, indicating this functionality stresses the idea that the user should be able to quickly transition between different levels of detail.
	Verify	By pressing the [Verify] button, the system should check business process diagram rules and indicate rule violations directly on the process canvas in the context of the process model. To demonstrate this functionality in the JIGSAW prototype, an error message is displayed when the [Verify] button is clicked.

Table 5-2: Functions for creating and manipulating a workflow.

5.3.3.5. Screen and Service Libraries

There are two types of libraries available to the user: a service library (see Figure 5-12) and a user interface screen library. Both libraries provide re-usable software components for workflow design. Services would most likely be web services which have been published in a UDDI directory (Universal description, discovery and integration). Screens could be iViews created using Visual Composer or other web user interfaces such as ASP.NET or JSP pages. Screens themselves need to be published in a centralized repository similar to the UDDI directory for web services before they can be re-used through the screen library in JIGSAW.

A library contains a column to the left containing categories of different services or UI screens. Once a category is selected, a list of services or UI screens is shown in the upper right section of the library panel. The lower right section shows the details of the currently selected components.

There are two ways to place a component from the library onto the workflow canvas:

- By pressing the [Add service] or [Add UI screen] button within the detailed component view in the lower right section of the library, a corresponding symbol is added to the workflow canvas.
- After clicking once onto a service (yellow) or screen (blue) in the upper right section of the library panel, a corresponding diagram symbol can be dragged across the canvas and can be placed into the workflow model.

The label of a diagram symbol can be changed by double-click once a symbol has been added to the workflow canvas.

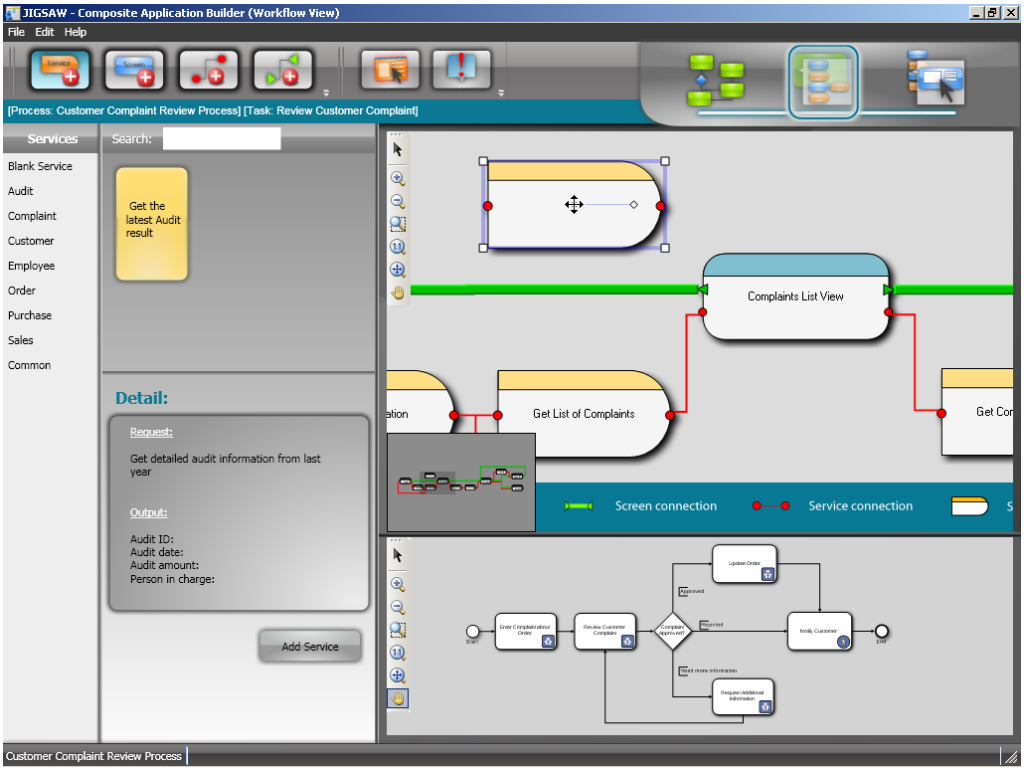


Figure 5-12: Library which offers re-usable components for workflow composition.

5.3.3.6. Verifying the workflow model

JIGSAW prototype demonstrates how an error in the workflow model should be indicated within the workflow canvas (see Figure 5-13). The verification should check the currently displayed workflow model against a set of rules to assure its consistency and integrity. Such rules might be checking syntactical correctness such as dangling connections, or might check semantic correctness by checking if input and output parameters of screens and services match. When the current workflow model violates rules, appropriate warnings or error indicators should be shown within the workflow model next to the diagram element which causes the rule violation.

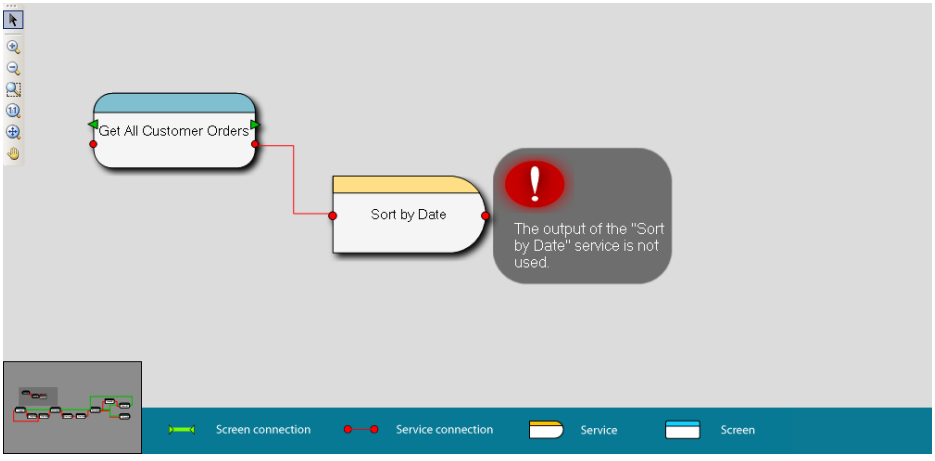


Figure 5-13: An error indication after verifying the workflow model.

5.3.4 Test View

5.3.4.1. Overview

Once a workflow design has been completed, the user can choose to test the workflow. The work area is again split into two halves: the upper half is used to visualize screens and service overviews of the workflow, while the workflow model is presented in the lower half of the work area (see Figure 5-14).

While the application is in workflow test mode, the user can interactively test the flow of screens and inspect and manipulate the behavior of services. The currently displayed screen or service preview in the test canvas is also marked red in the workflow diagram. While testing, the user can analyze data dependencies between workflow entities. Once a test has been completed, a test report is displayed in the test canvas summarizing data entry, service call results and time spent per workflow step.

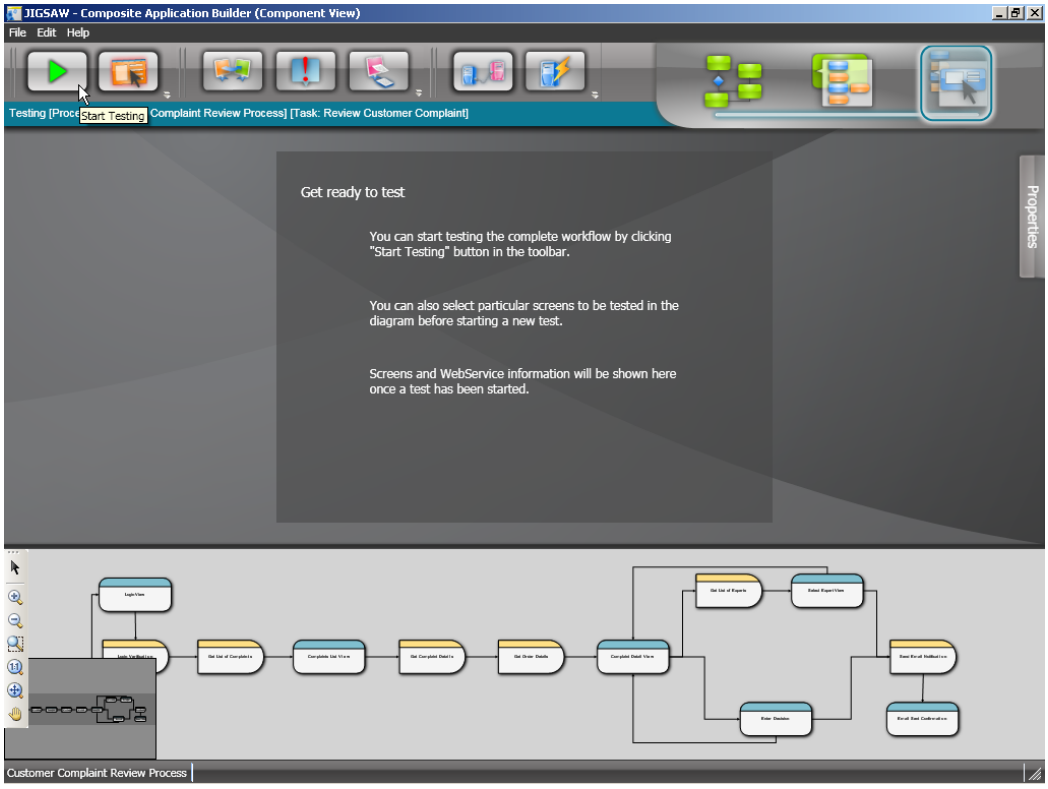


Figure 5-14: The Test View

5.3.4.2. Main Toolbar

The main toolbar (see Figure 5-14) contains functions to start and stop a test, to edit screens, to share a test case with stakeholders, to verify screens and services, to publish screen and service details to a document (e.g. Microsoft Word document which makes inline editing easy), to visualize data dependencies and to activate service calls. Refer to Table 5-3 for a detailed description of these functions.

Icon	Function Name	Interaction
	Start and Stop a test	Click to start or stop the test. This is a toggle button which shows a green play icon when

		a test can be started and a red stop symbol when a test is running.
	Edit	Not yet implemented. Should allow a user to change into a screen edit mode while inspecting screens before a test.
	Share	Not yet implemented. Should publish a test so that it can be run by other stakeholders or even end-users. This functionality is important when end-users need to buy-in to changes of a workflow.
	Verify	Not yet implemented. Should allow a user to verify the integrity and consistency of screens and services before a test is started. This verification should also be integrated with the intended data dependency visualization.
	Publish	Not yet implemented. Should allow a user to publish a workflow test as a document which can be published or shared among stakeholders. We recommend the generation of Microsoft Word document format for quick inline editing.
	Show data dependencies	Not yet implemented. Should allow a user to inspect the data dependency between screens and services before a test is started. The data dependency view should visualize how parameters are mapped in data flow connections between services and screens.
	Activate and Deactivate service calls	Shows a dialog (see Figure 5-15) which allows a user to activate or deactivate service calls before performing a test.

Table 5-3: Functions for testing a workflow design.

5.3.4.3. Service Call Activation

By deactivating a service call, the user can circumvent run-time errors caused by services which are not available for testing. Users requested during prototype think aloud tests that they would like to deactivate service calls which involve long running transactions (transactions which take more than a few seconds to terminate).

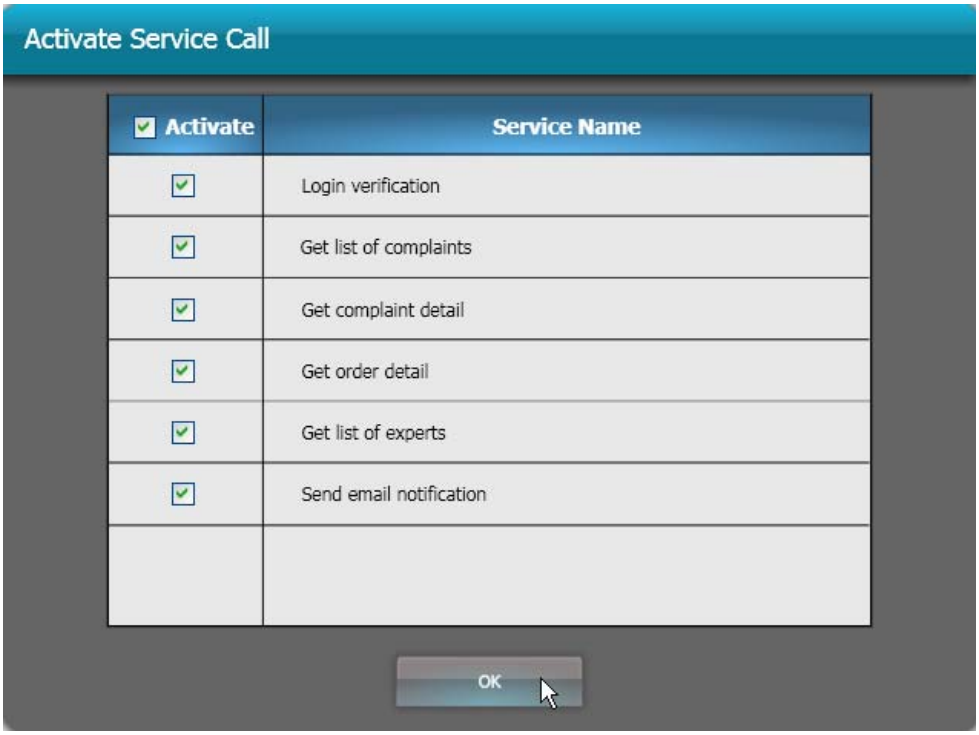


Figure 5-15: Service Activation Dialog

5.3.4.4. Service Visualization

While a test is in progress, services are visualized for inspection of input and output parameters (see Figure 5-16). Services are visualized using the shape and color introduced for the workflow canvas symbols. If a service has been activated through the service activation dialog, the user can execute the service call by pressing the [Execute] button in the header of the service. Once a service has been called, the output parameters are displayed. If the service call caused an error, an error message with detailed error information is shown (see Figure 5-16).

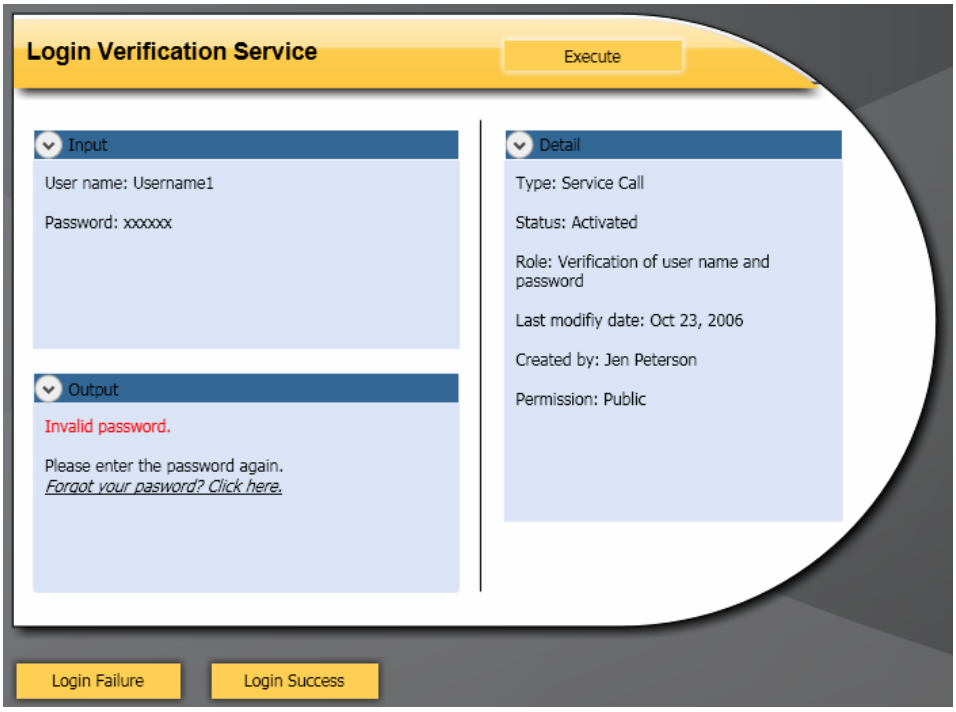


Figure 5-16: Visualization of a service during test.

5.3.4.5. Test Report

At the end of a workflow test, the system provides a test log listing the tested screens and services including the parameters which have been processed (see Figure 5-17). The test log provides a summary with a time stamp referring to the test start time, with the number of screens and services tested, and the total time spent testing.

Each screen interaction and service call is separately listed, showing input and output parameters and the time needed to complete the interaction or the service call. However, the time measured for user interaction is not a reliable indicator for user performance. Instead, we envision that a workflow designer might want to keep a record of logs to track how workflows and their tests change over time. This log can be an indicator for skilled user performance when captured during tests with end-users who are knowledgeable about the workflow.

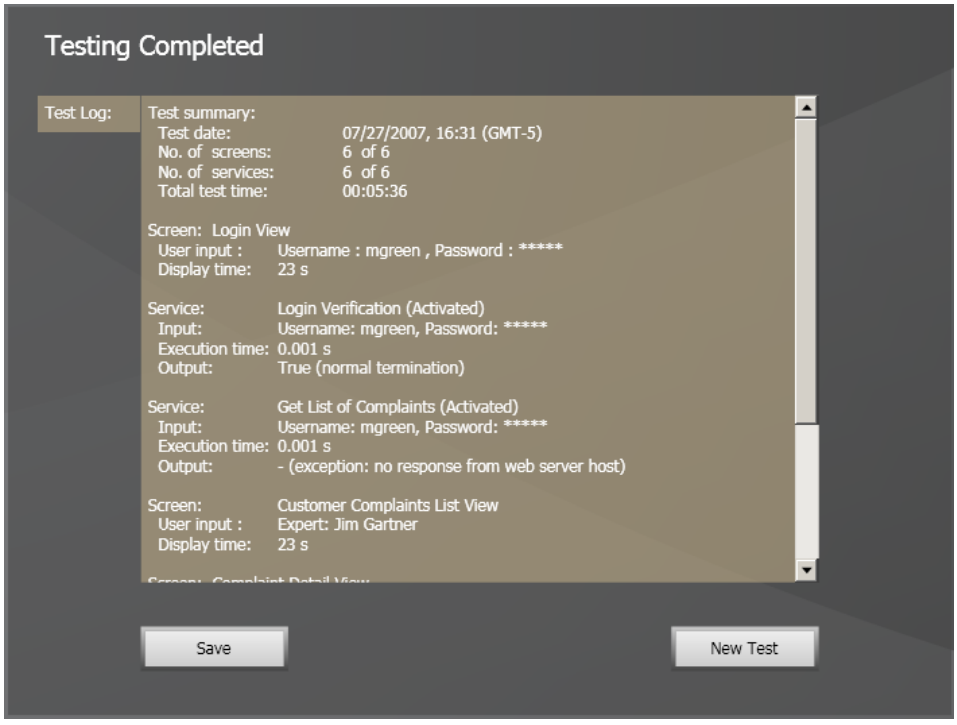


Figure 5-17: Test Report shown at the end of a test run.

5.4 Future Work on Prototype

5.4.1 Contextual help

Contextual help has been determined as important during concept validation. We therefore suggest making both menu bar and status bar contextually aware of the user's current task. The menu bar should provide functionality depending on the current application mode. The status bar should provide help relevant to the user's current action. Both the menu bar and the status bar could be vital for providing contextual help throughout the JIGSAW application.

5.4.2 Keyboard Control for Canvas navigation

In order to achieve flexibility and efficiency of use, the design canvas should allow keyboard input to move, re-name and delete selected elements [13]. Deletion of selected elements has been implemented for both the Process Canvas and the Workflow Canvas. However, keyboard control to move selected items should be implemented as well.

5.4.3 Testing Options

Before a test starts, the user has the option to activate and deactivate service calls. However, the user should also have the option to only test screens of the workflow, while service details remain hidden.

5.4.4 Testing a process including multiple workflows

The JIGSAW prototype demonstrates how a workflow can be designed and immediately tested. However, users also expressed the need to test whole processes including multiple workflows. They would like to impersonate different user roles and verify the consistency and integrity of workflows which are executed in sequence.

5.4.5 Designing variations of workflows for different user roles

Workflows and thus composite applications are often executed by multiple users with different roles. In JIGSAW prototype, roles of users can be assigned in the property in the test view. However, workflows might vary depending on the user's role. The prototype should be expanded so that roles can be permitted to execute certain screens or services in the workflow. A test should then consider the selected user role to show or hide particular screens or services at test run-time.

6 Design Process

In this section, we will walk through all the intermediate steps that led us to our final design and prototype. For each step, we will discuss the methods we applied, findings that refined our solution, challenges we encountered and our learning. This section also documents the design ideas we experimented, but decided to exclude from the final solution due to time and scope constraints. These ideas may serve as the basis for future work on our prototype.

6.1 Brainstorming Session

Post user research, our team went into the generative phase. We first analyzed findings from our contextual design models and used them as the basis for new design ideas. Each team member was given the opportunity to develop and explain his or her own concepts to the entire team. The team then as a whole discussed, evaluated, and augmented each individual idea. In order to keep the creative level high, it was important that each member freely expressed his or her ideas. These ideas were the stepping-stones to a well-tested and robust design.

6.2 Concept Validation

Loaded with design ideas, we proceeded to develop a set of forty-one (41) different storyboards to further verbalize our initial ideas and seek validation from users. The full set of concept storyboards can be found in Appendix C.

6.2.1 Methodology

- Explain to the users that the purpose of concept validation is to match user perceived needs with observed needs
- Each storyboard is comprised of a scenario depicting the use of potential composite application design-time solutions or features, lead questions that aimed to probe the user need, and discussion questions that focused on elaborating the form of the solution (Figure 6-1).

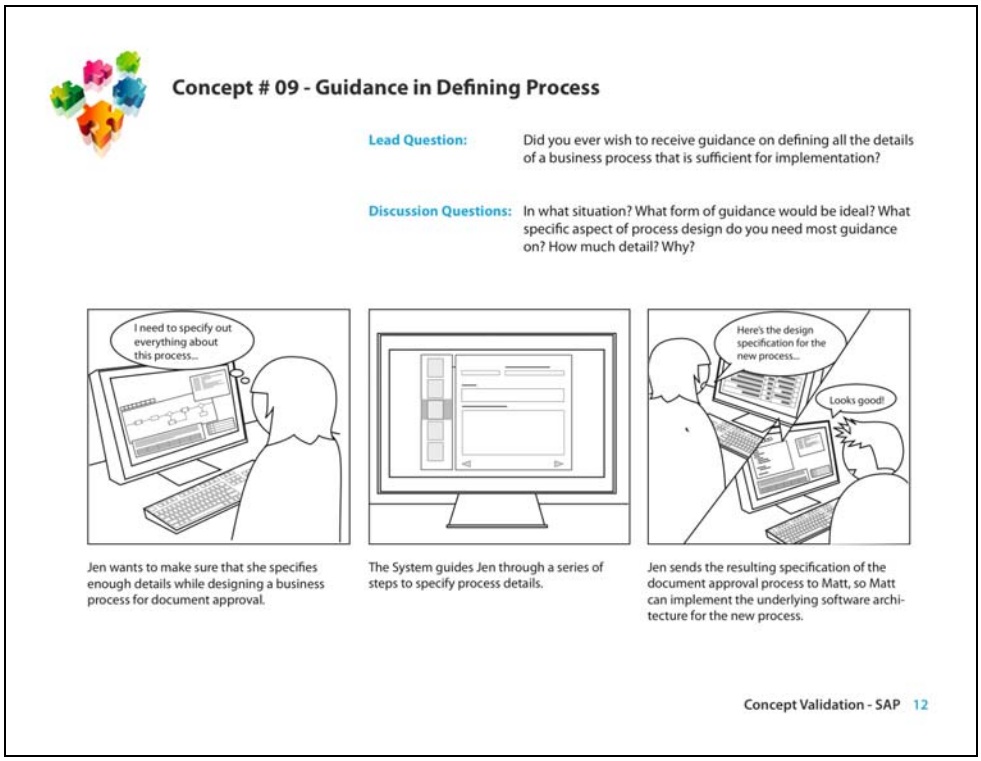


Figure 6-1 A concept validation storyboard contains a short scenario, lead question, and discussion questions. The storyboard is used to find the overlap between the needs that the researchesr observed and needs that the users perceived.

6.2.2 Concepts

These storyboards covered a rich variety of our design ideas that were inspired by the design implications derived during the research phase, and can be generalized into the following categories:

- Improving business process comprehensiveness
- Enhancing the awareness of business process requirements
- Easing the modeling of composite application
- Increasing the robustness of composite application design
- Improving composite application life-cycle management
- Making collaboration among members of cross-functional teams more efficient

6.2.3 Validation Sessions

To validate the needs we observed during our user research, seven (7) individual concept validation sessions were held with business process experts. This method ensured that the most important user needs were captured and addressed by the solution we proposed. During each session, we went through the storyboard scenario, lead questions, and discussion questions with the participant.

Due to the time limitations, we had with each validation session participant, it was impossible for us to discuss all 41 concepts in one session. Therefore, we handpicked 10 concepts to present for each session. During the handpicking process, we

prioritized some concepts to be shown to more users based on the team's collective interest while ensuring that each concept was validated by at least one user.

Most of the concepts were well received. The collection of direct feedback from users can be found in Appendix C.

6.3 Setting Design Foci

After our concept validation session, the team was faced with the challenge of transforming forty-one (41) diverse concepts into one single coherent system. In order to help us move forward, we plotted out all the concepts along the timeline of a business process lifecycle and attempted to identify those that would comprise an end-to-end solution.

However, we noticed that most of the best received ideas fell around the later part of the business process design phase, where the process gets modeled with concrete specifications and is evaluated against the business users' needs. We decided that there is a great opportunity in designing a composite application design-time tool that integrates with the user's current practice of business process modeling and testing.

6.3.1 Takeaways

We prioritized a number of high-level key features in our system that were extracted from our original forty-one (41) concepts:

- Visualization of composite application design via business process modeling notation (BPMN) and user interface thumbnails that users are familiar with.
- Zoomable user interface that provides smooth transition between the high level process view and detailed workflow view
- Search-enabled library of web-services, components, and pre-built workflow that fosters reuse of best practices. The library is further backed by an online community, where people share experiences and best practices.
- Testing environment that simulates the proposed composite application in action and generates a prototype of the application that users can play with.
- In-model annotation that can be used in automatic documentation generation.
- Documentation organizer that can be linked to a part of the model
- Embedded business requirement approval system
- Visualization of dependencies between different parts of the workflow as well as inter-process dependencies.
- Contextual help that guides users in specifying the necessary implementation details of a composite application.

6.4 Wireframes: Iteration 0 (P0)

In order to shape our chosen features into a concrete definition of the system, we first had each individual team member sketch some low fidelity wireframes for the key features mentioned in previous section. This method allowed us to provide a common ground in discussing how all the different features can be pulled into one system and act together coherently.

6.4.1 Features

Below are all the features we attempted to wireframe. The actual wireframe illustrations of individual team members can be found in Appendix D.

- Constructing flowchart diagram using a freestyle diagramming paradigm.
- Reusing an existing process from the process library that connects to an online business process expert (BPX) community.
- Visualizing the workflow with user interface flowcharts.
- Approving business requirements associated with the business process model.
- Tracking and managing business process project timelines.
- Navigating the business process diagram using Google Earth style controls.
- Changing granularity of the visualization with slider.
- Visualizing the dependency between different processes over a process flow diagram.
- Attaching annotating documentation to a process flow diagram with a color-coding schema.
- Composing a workflow demo in video format.

6.5 Setting Prototype Foci

Through wireframing, we realized that each key feature is extremely rich in its very own design space. Given the time we had for this project, it was inevitable we would have to further prune down the scope of our design.

6.5.1 Feasibility-Novelty Matrix

Our team leveraged the method of cost-value matrix to help evaluate the various features. In order to tailor to our project goal given by our client, we replaced cost and value with feasibility in terms of both prototype implementation and user acceptance and novelty of the idea. The evaluation process involved three steps:

- Each team member spatially placed each key feature in the form of a sticky note onto the feasibility-novelty matrix based on his or her own perception.
- The team averaged the placement of all the sticky notes representing the same feature.
- The team selected the features that are in the quadrant of high feasibility and high novelty.

Figure 6-2 shows the resulting matrix.

Figure 6-2. The feasibility-novelty matrix. Blue sticky notes are individual placement; yellow sticky notes are consolidated placement.

6.5.2 Takeaways

After taking into consideration the interests of our client, we came to the following three foci for our prototype:

- Workflow visualization
- Zoomable user interface
- Testing environment and prototype delivery

6.6 Low-Fidelity Prototype: Iteration 1 (P1)

In our first design iteration, we used use case scenarios to detail the paper prototype. A total of seventeen (17) scenarios were generated. Using these scenarios as a basis, we constructed a series of screen steps animating how the system would react to user interaction (see Appendix E).

6.6.1 Prototype and Design Rationale

6.6.1.1. Composite Application Composer

Our first prototype (P1) tested the idea of modeling composite applications using two-level flow diagram visualization, which was later adapted into our final design. Figure 6-3 illustrates the notations we used.

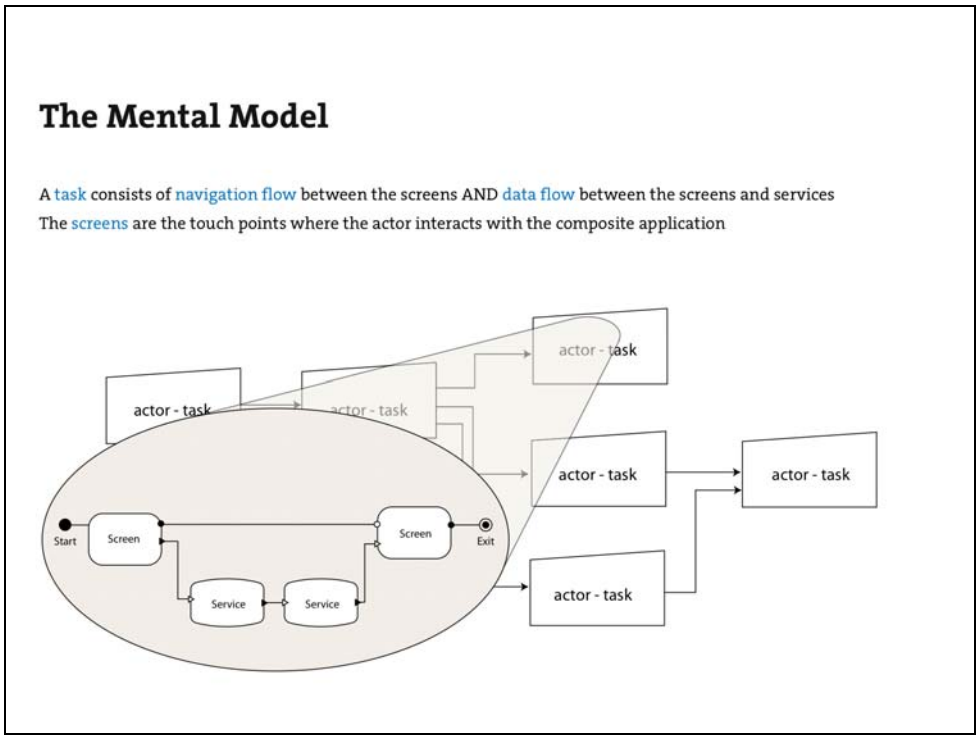


Figure 6-3: We simplified and customized a set of notations to form a visual representation of the composite application in P1.

For this iteration, we proposed a simplified paradigm for business process modeling to reflect the fact that each element in the process flow diagram is essentially a container for a workflow. A process was merely a sequence of tasks, and a high-level step was taken by one actor independently.

The multi-level workspace also naturally incorporates the usage of pre-built templates (Figure 6-4). This prototype presented users the choice of templates available whenever the user wants to create a new workflow or a new screen. This feature later evolved to the library feature in the final design.

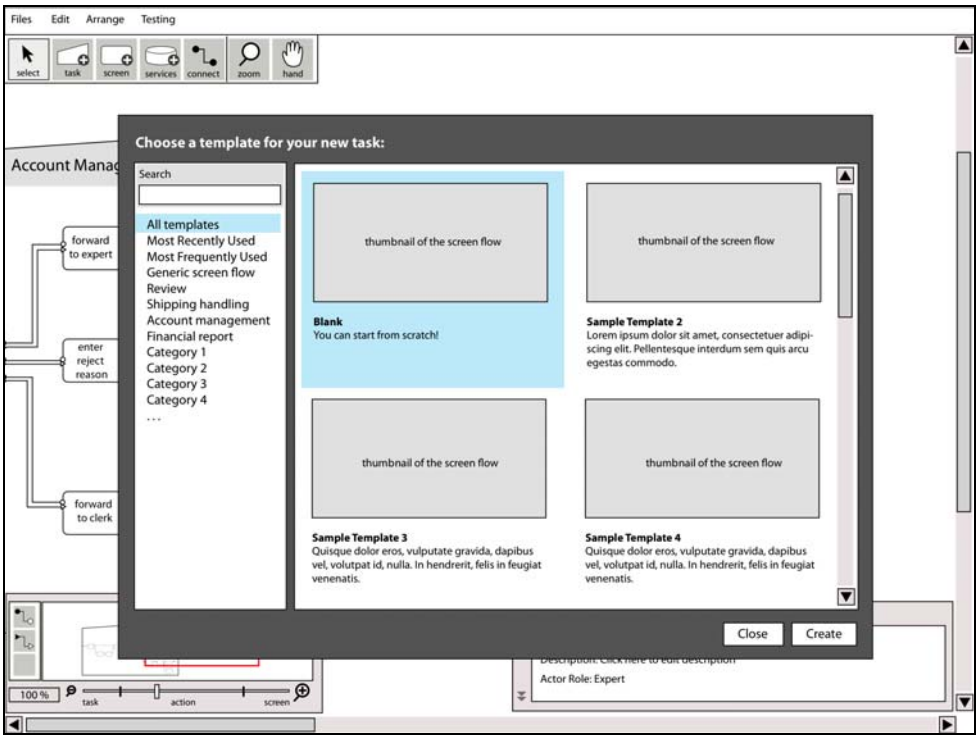


Figure 6-4: The template chooser allows users to browse or search through a huge library of pre-built best practice templates as the basis of a new workflow model. This feature evolved into the concept of library in our final design.

The concept of zoomable user interface (ZUI) and multi-granularity view of the process was also introduced in this version. A user could go from the process level to workflow level and from workflow level to user interface level by zooming (or enlarging) into a task and further into a screen in a task's workflow. Similarly, to come back out to the process level from user interface level, a user could zoom out using the zooming level controls provided. Since ZUI provides smooth transition between different contexts while maintaining the spatial orientation of the process visualization, we believed that it would be a compelling solution for naturally switching between the three different editing levels. Figure 6-5 shows the zooming controls we proposed in this prototype.

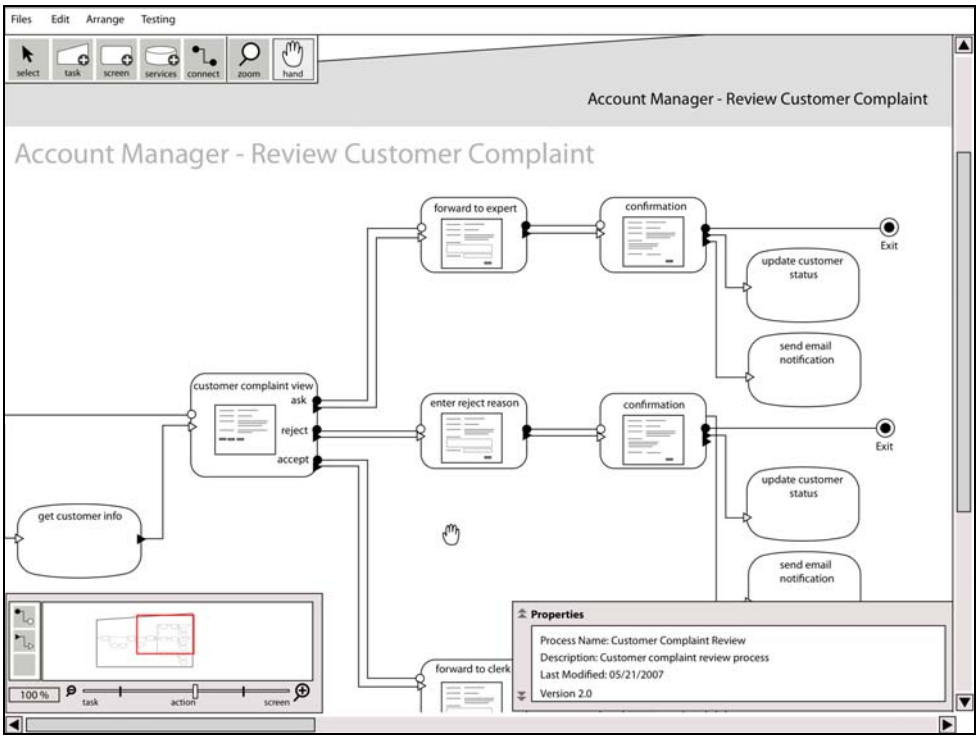


Figure 6-5: In P1, users can use the slider at the lower left corner of the canvas to zoom-in or zoom-out. There are also navigation tools in the toolbar at the top of the canvas.

6.6.1.2. Simulation Environment

Another component we tested in this design iteration was the concept of business process simulation, which outputs an interactive prototype of the business process. Figure 6-6 shows the screenshot of the simulation environment we proposed.

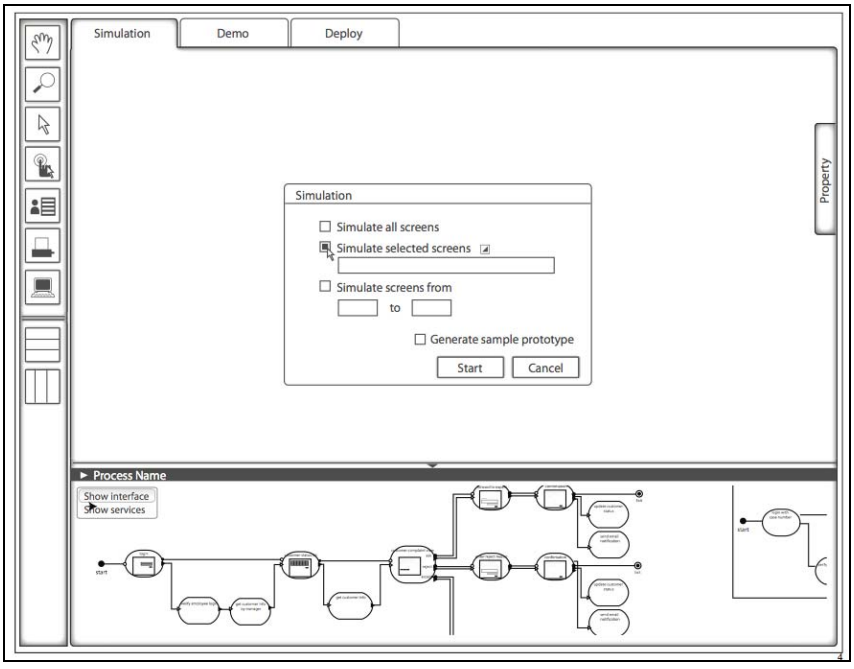


Figure 6-6: Simulation environment in P1

Due to the complexity of business processes and the various mechanisms for business process design, process simulation can be very time consuming. We found it

was essential to not only enable testing the whole process, but also individual parts of the process.

Therefore, in this prototype, we provided both styles of testing. Users could either choose to simulate the entire process or choose a section of a decision path in the process model to simulate. We forced users to choose a path instead of random steps to prevent errors caused by disconnected data input and output from services and screens.

We also proposed a number system for the workflow, in which each screen and service was assigned a unique number. Figure 6-7 shows the number system in workflow selection. Users could use these numbers to identify the simulation path.

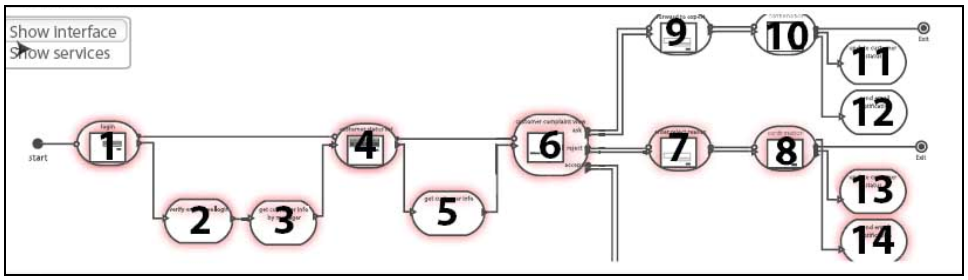


Figure 6-7: Simulation environment - numbering system

The ability to detect flaws in the process integrity was also incorporated in this iteration. The errors were highlighted in different colors in the workflow canvas of the testing environment. Moreover, to make the workspace more flexible, once a flaw is detected; users can easily switch to the modeling environment to edit the workflow. Figure 6-8 displays a close view of this function.

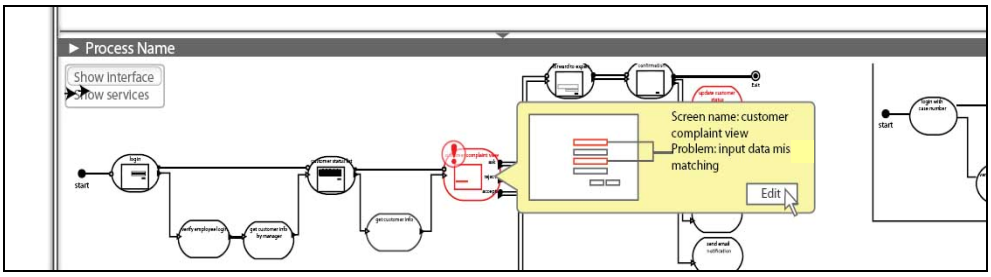


Figure 6-8. Simulation environment - flaw detection

The interaction with the prototype impersonates real web applications; the data users input will be passed along to the next screen or service. A user is able to interact with the screens only. We proposed in this iteration that the simulation environment would alert users for invalid input based on the data type of the input field. We believed that it would help users to catch runtime errors and exceptions.

Our user studies brought up the possibility that some essential service calls cannot be completed during the testing session due to a broken or inactive service (see Appendix E for details). To handle this concern, which most likely would impede a successful testing, this iteration allowed users to enter customized data for missing services calls. Moreover, we also designed that this prototype would save this customized data and pass them along to help the user repeat testing. Figure 6-9 gives a close view of this feature in P1.

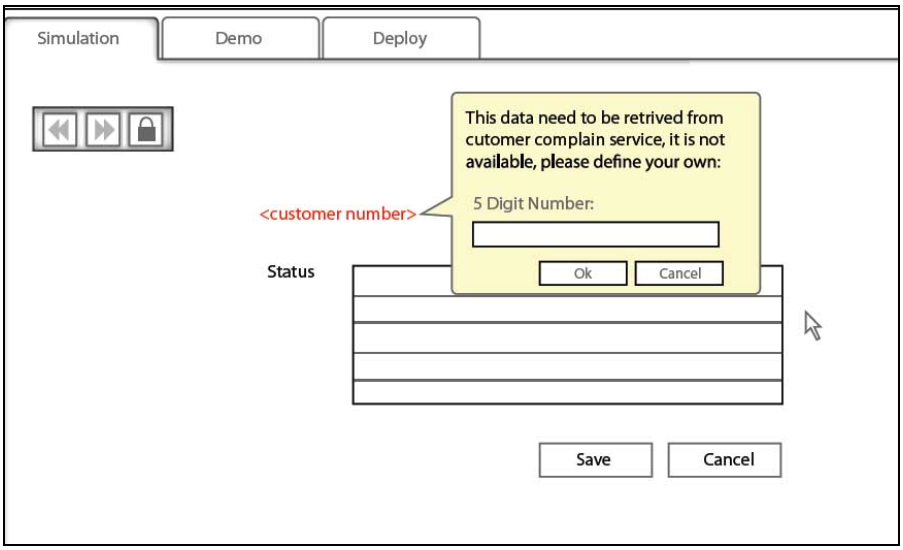


Figure 6-9. Simulation environment - input customized service output.

This round of prototyping also tested the concept of the service dependency. Figure 6-10 shows the visualization of service dependency.

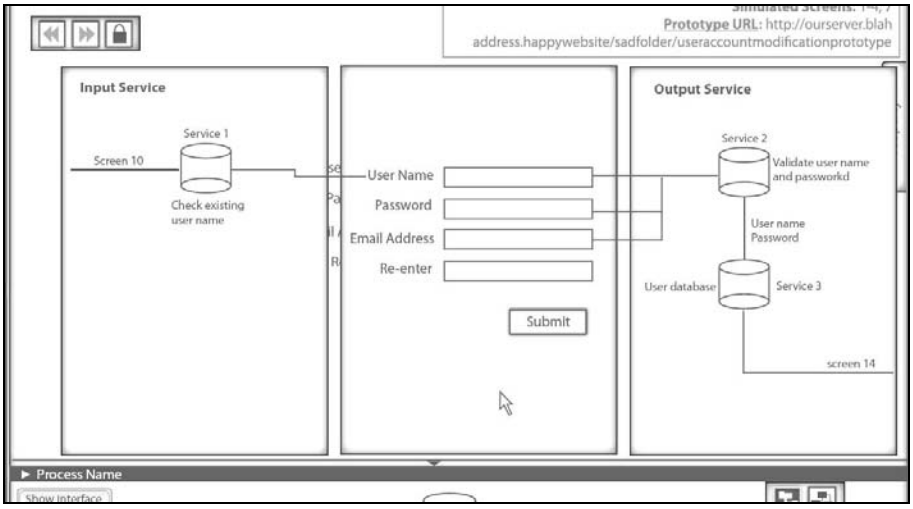


Figure 6-10. Simulation environment - service dependency

We also suggested the concept that a prototype can be published, in which the system provides a simple URL for business process experts to pass around (Figure 6-11).

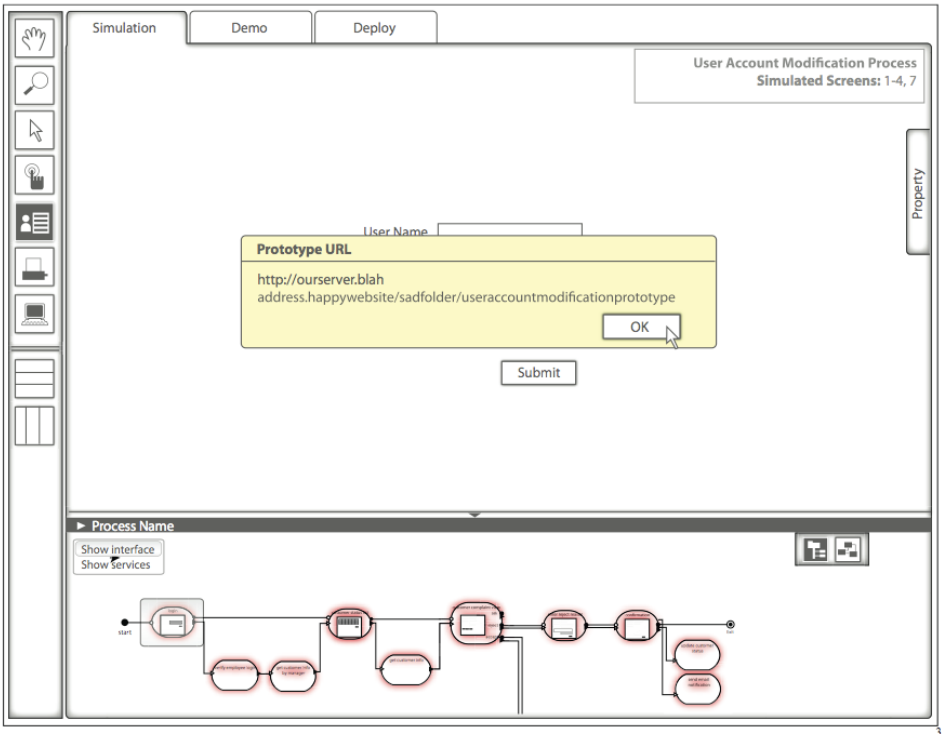


Figure 6-11: Simulation environment - publishing prototype.

6.6.1.3. Demo Video Studio

The last component of this iteration was a demo video creation and editing tool for the newly created composite application. This demo video studio consisted of two parts; the first part enabled users to capture mouse movement over application screen and to record audio narration; the second part provided basic video editing and annotation functionalities. Figure 6-12 shows the screenshots of demo video studio.

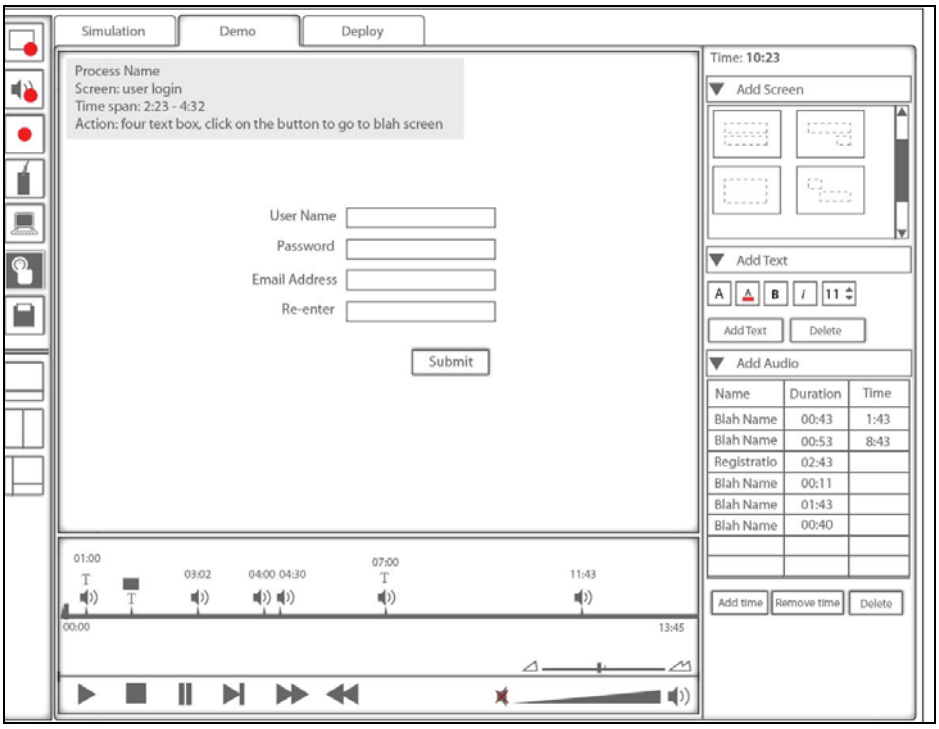


Figure 6-12. Demo video studio

From user research, multiple business process experts complained that making a demo video for processes is a complex and painful process, mostly because audio and screen movement recording tools are usually separate (see Appendix E for details). In this iteration, we combined both audio and screen recording tools so users could record the narration while recording the screen movement.

Many advanced video editing tools are considered hard to use for non-expert users. Our concept was to simplify the user interface, eliminate many of the advanced video editing features, and provide basic tools that serve the purpose of creating a demo video. We believed that a simple interface with only the necessary functions would help users efficiently in creating deliverables that can be presented to the .

6.6.2 User Testing

We performed Think Aloud Studies with ten (10) business process experts to evaluate this iteration. During the study, users were first introduced to the two-level flow diagram visualization scheme. Users were then asked to complete several tasks framed in the use case scenarios and articulate their thought processes as they went through the tasks.

Normally such user testing session would be conducted in person. However, due to the challenge we faced in participant recruitment, we were only able to talk to most of the users remotely through teleconference. In order to accommodate the situation, we shared digital copies of our designs with the users. We then directed them to jump to a certain page as a reaction to their proclaimed interaction with the prototype. We also constantly encouraged the users to think aloud and probed them to give a precise description of the interaction they would take to compensate the fact that we were not able to observe the users' actual actions. One of the drawbacks, however, is that the screens we prepared could not possibly catch all the combinations of different system states. In those cases, we verbally described the

system response using common metaphors and then followed up on the users' reactions.

For this prototype testing round, we were particularly interested in seeking answers to the following questions:

- Can users associate business process modeling with composite application modeling?
- Do users understand the two-layer flowchart paradigm?
- Are users comfortable with the zooming interaction? With switching workspace environment using the zooming interaction?
- Will users value the usage of the templates?
- What is the user trying to test?
- Does the user ever want to select part of the workflow to simulate?
- Is there a need to show service dependency?
- Does the user ever wish to skip simulating some screen while in simulation mode?
- How does the user want to publish the prototype?
- Is the demo video studio helpful?

Additionally, we asked users about desired functions, understanding of the toolset usage, language used, and any other relevant feedback. The complete collection of direct verbal feedback from the user testing sessions can be found in Appendix E.

6.6.3 Takeaways

The most important observations we learned from user testing are summarized below. These insights also became the basis of the rationale for our next design iteration.

6.6.3.1. Flowchart visualization and workflow modeling

One of the main observations from the user testing was that users could not tell the difference between modeling a process from modeling an automated workflow. We received a number of comments on missing functionalities (e.g. how to model a step where no system is involved, how to assign an actor to a single screen element) while users were actually performing tasks related to automated workflow modeling. It was evident that as soon as users perceived a flowchart that resembles some kind of simple business process, users immediately associated the task with normal business process modeling, despite the fact that we introduced the differences between the two sets of notations beforehand and reiterated the task goal was to model a composite application to support workflow automation.

We were not be able to make the conclusion that users are not able to associate flowchart with composite application modeling due to the shortcomings of the Think Aloud methods and imperfection in paper prototype. Yet it served as a good implication that we should seek an alternative visual representation.

6.6.3.2. Unconventional flowchart elements

A few users suggested that they would like to see the conventional flowchart elements in the process modeling, especially the diamond for decision split and the

start and end terminal. Users also reflected that the shapes of service elements and screen elements were not distinctive enough to differentiate in a glance.

6.6.3.3. Navigation and zoom controls

A mixed reaction to the navigation and zoom controls emerged from the user testing. While a few users quickly recognized the zoom buttons and the usage of the slider, the rest were not able to see any affordance for zooming-in or zooming-out. The need of a "zoomed-out" effect on the diagram was, however, expressed universally.

More fundamentally, the majority of users demonstrated unfamiliarity with the drawing tools palette paradigm (i.e. toggling between select tool, zoom-in tool, zoom-out tool, etc.). These users expected to see immediate changes or feedback in the system environment rather than a change in the mouse behavior. Such an expectation was understandable given that most of business users are more familiar with the Microsoft Office applications in which the concept of tools palettes is not prevalent.

6.6.3.4. “Simulation” vs. “testing”

One of the main issues arose from the user testing was the mismatch between the users’ mental model of “simulation” and ours. The “simulation” we proposed that was a step to produce an interactive prototype, which simulates how the composite application would work. However, “simulation” in business process experts mind often refers to process simulation, which involves imitating how the business will run with the to-be process and evaluating resource bottlenecks. Many user study participants suggested that the functionalities we described could be better encapsulated as “testing.”

6.6.3.5. Workflow selection

From user testing, we perceived some usability problems the way partial workflow selection was designed. Most users were clueless in how to select a path that contains all the related screen or service. Some users suggested following the route selection mechanism in which choices of routes are displayed as decision point is reached.

6.6.3.6. Service activation

There was a mixed reaction to service call activation feature in user testing. Some users said that they would prefer to disable service call at all times, and always input customized input data. Others preferred activate the service call at all times, to test whether the call would be executed successful. Many users valued real life interaction with service calls activated at all times, so they are able to see how the process would behave before it is deployed.

Additionally, users raised concerns about the case when the services are known to be broken or unavailable. Users suggested having the option to choose which service they want to be activated.

6.6.3.7. Setting simulation environment variables

A few users pointed out that sometimes there is a need to set the simulation environment variables such as times and user roles. Some services may provide very different result at the beginning of the month comparing to the end of the month; some services may only be accessible given enough user permission.

6.6.3.8. Export user interface screens

The participants of our user studies were not very excited about the concept of demo video studio in general. One user expressed that tools as such already exist and are in use. On the other hand, the participants recommended that, rather than a video, there might be more use in having screenshots of the user interface of the tested composite application. These exported screenshots can be further used in documentation of the process as well as in creating a tutorial material.

6.6.3.9. Navigation freedom

The most common suggestion from users was to increase user freedom during simulation, such as jumping ahead to test other screens or services in the workflow. Moreover, users should be able to input any customized data in both service and screen at any time.

As an additional note, a few users expressed that the most annoying thing in a zoomable user interface was not being able to quickly reposition the canvas back to the previous view if the view was changed. The users further suggested that it would be nice to have some easy way to return to a certain view, such as a back button to go back to previous view or one-click access to the zoom-level that can view the entire process flow.

6.7 Low-Fidelity Prototype: Iteration 2 (P2)

To further experiment with the idea of workflow modeling visualization, we proposed an alternative design on the workflow modeling interaction. We chose to stay at the level of low-fidelity paper prototype to rapidly validate our design idea. We also elected to cut back on the demo video feature and to shift our effort away from designing a robust set of process flowchart editing capabilities to concentrate on the other features that were deemed as more innovative and important by both the users and our client. As a result, ten (10) new use case scenarios were outlined for this iteration (see Appendix F for use cases).

6.7.1 Prototype and Design Rationale

6.7.1.1. Composite Application Composer

User testing had revealed that our design of workflow modeling visualization required more improvements to be easily understood by users. In this iteration, we moved away from the flowchart paradigm and leveraged the workflow visualization similar to Apple Automator [4] (Figure 6-13). A workflow, in the design of this prototype, consists of sequence of actions in the order they would be triggered.

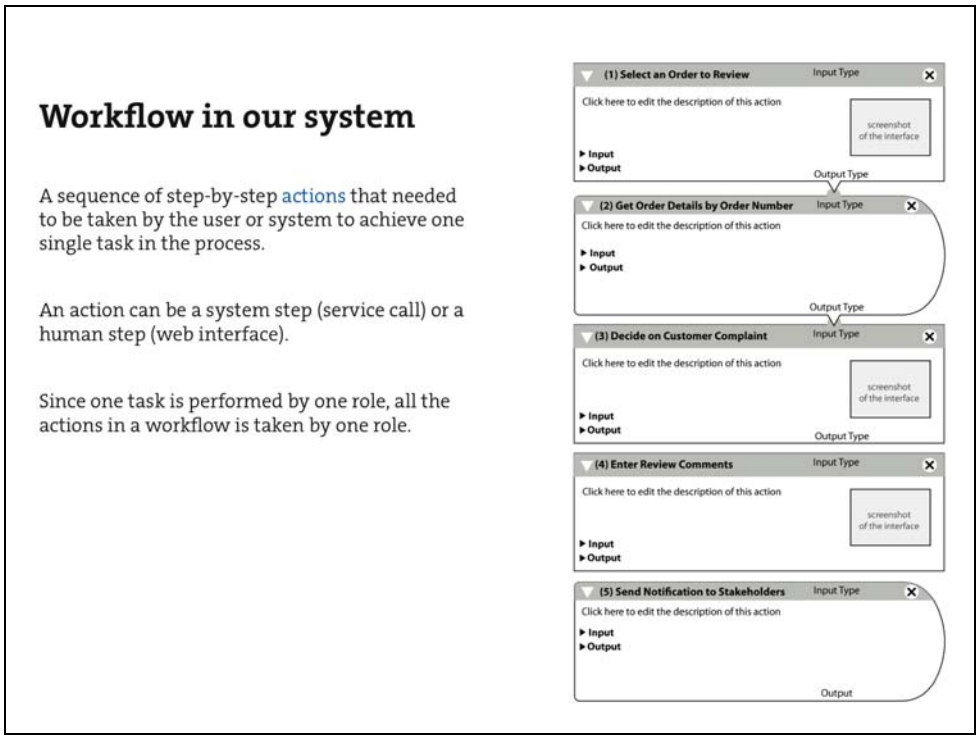


Figure 6-13. This iteration leveraged the Apple Automator visualization of a workflow. The square action element represents a human step (e.g. web user interface, Adobe interactive forms, etc.). The action with round side represents a system step (e.g. web service call).

Beyond the drastic difference in orientation (horizontal for process, vertical for workflow), this new representation offered a more rigid and straightforward construction of workflow, as opposed to the freestyle box-and-line representation. This visualization waives the affordance in creating over-complex workflow structures, which was considered as "bad practice". It was our intention that users should structure the high-level business process in a way such that each workflow model remains independent and maintainable as dictated best practice for composite application design.

Adding a new step to the workflow in this iteration also changed dramatically in comparison to previous iteration. With the Action Library (Figure 6-14, see Appendix F), all the possible workflow augmentations were unified into one single series of simple interactions, namely search, select, and then drag-and-drop.

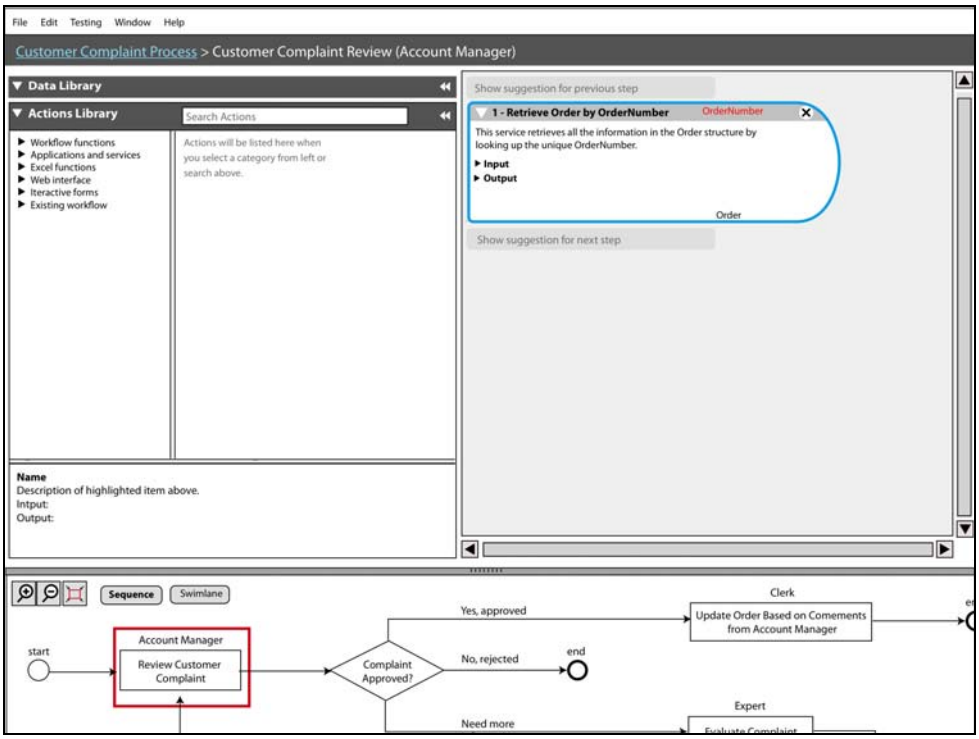


Figure 6-14. The Action Library (middle left) contains all the possible elements users would ever want to add to their workflows.

The Action Library integrated better with the workspace environment and provided better scalability. In previous design, the components were forced to be categorized into either screens or services; anything that did not fit would require adding another button to the toolbar. We also observed from previous iteration's user testing that the pop-out template chooser caused minor disruption to the user's task flow. In this iteration, the Action Library allowed different actions to be listed in the categories that best describe them. With the entire library embedded within the workspace, users could carry out the task more smoothly and be aware of all the available options.

We also proposed that the users could get suggestions on which action is suitable to be inserted. We imagined the suggestion could be based on matching data I/O type, common combinations, and individual usage patterns. This feature serves as a more proactive solution to reduce the selection barrier [3].

Defining data mapping perhaps is the most technical part in the task of constructing a comprehensive composite application model. Another concept we introduced in this iteration was the Data Library (Figure 6-15). The Data Library acted as a central registry of all the data output of all actions in the workflow. When a user wants to define the source of the input to an action, the user could quickly view all possible mappings pre-filtered by the data type and apply the mapping by drag-and-drop. There were a number of benefits in this design:

- The technical complication of data scope and binding is hidden from the business users.
- Users do not need to examine each action in the workflow to look for the right source. Users also no longer need to "reach" across a long distance to make the connection if the source action and the destined action are far apart.

- Users can be aware of all the possible candidates when making decisions on which source to use. Users can also easily examine the data dependency by looking up all data mappings related to one source.
- By avoiding using lines to represent the data flow in the diagram, the canvas would not be cluttered with too many lines.

Additionally, we imagined that the system could perform data mapping automatically to a certain degree based on matching field type and name.

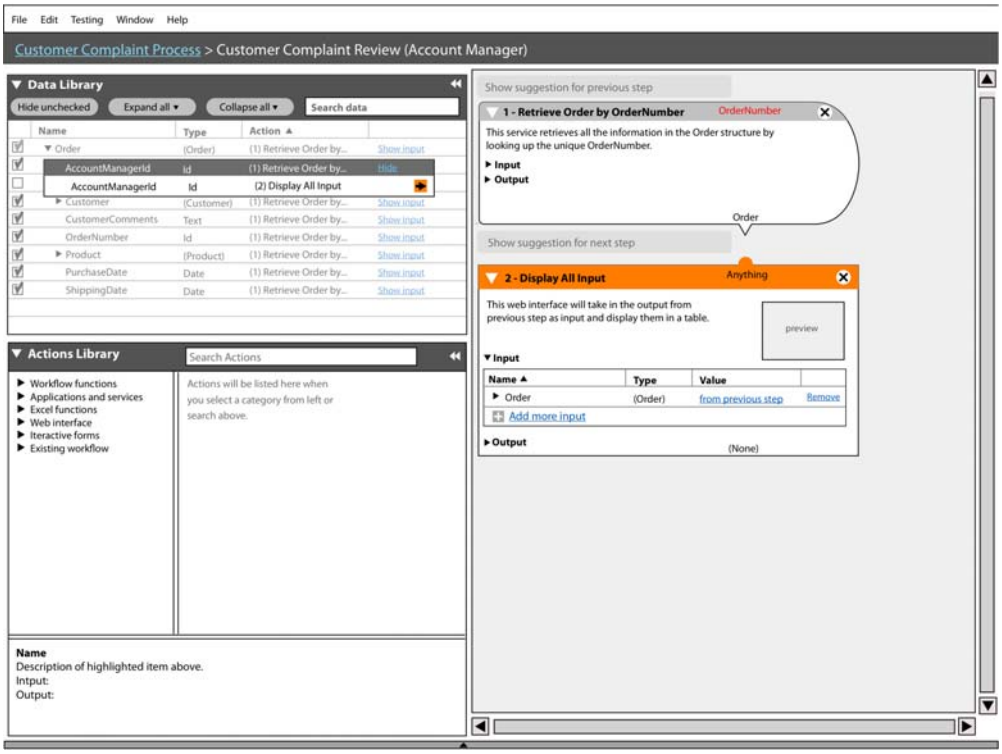


Figure 6-15. The Data Library (upper left) serves as a central data registry and the main point for defining data mappings throughout a workflow model.

Finally, we replaced the zoom controls that were controversial in previous iterations with the Google Map navigation/zoom controls (Figure 6-16) to experiment with user acceptance in the context of flowchart editing. We also added a Navigation Path bar to allow single-click repositioning due to popular request during the user testing.

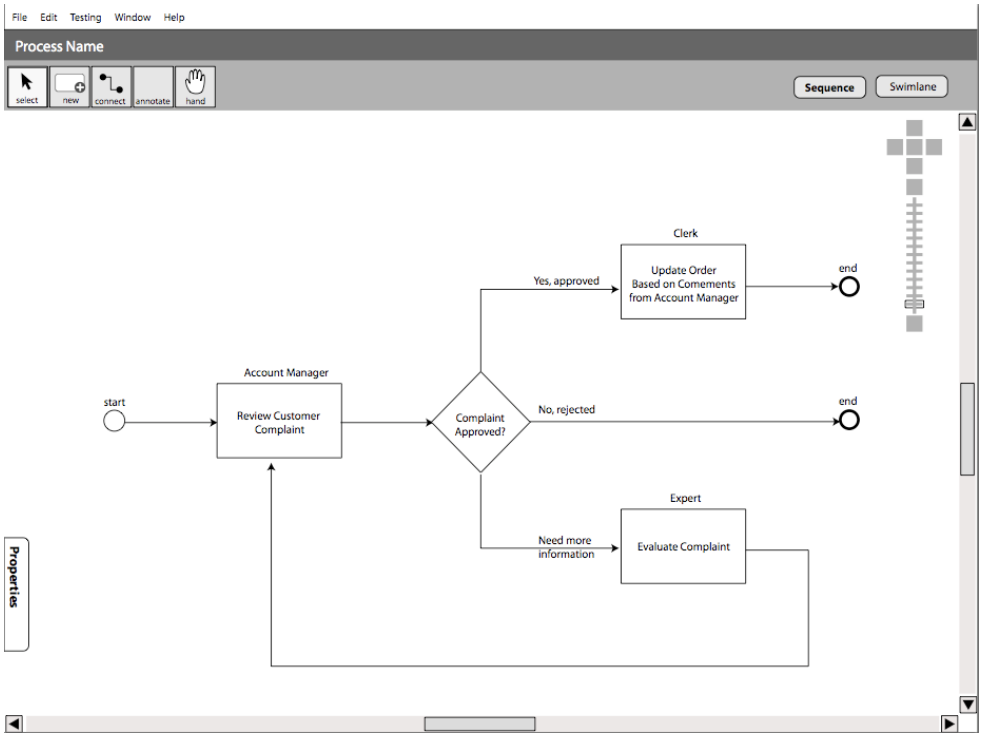


Figure 6-16: The process modeling canvas can be navigated with the GoogleMap-liked controls.

6.7.1.2. Testing Environment

Based on the feedbacks from previous design iteration, we rephrased "simulation" to "testing". Once the user is in the testing environment, a start-testing button could be used to trigger the interaction between users and screens or services.

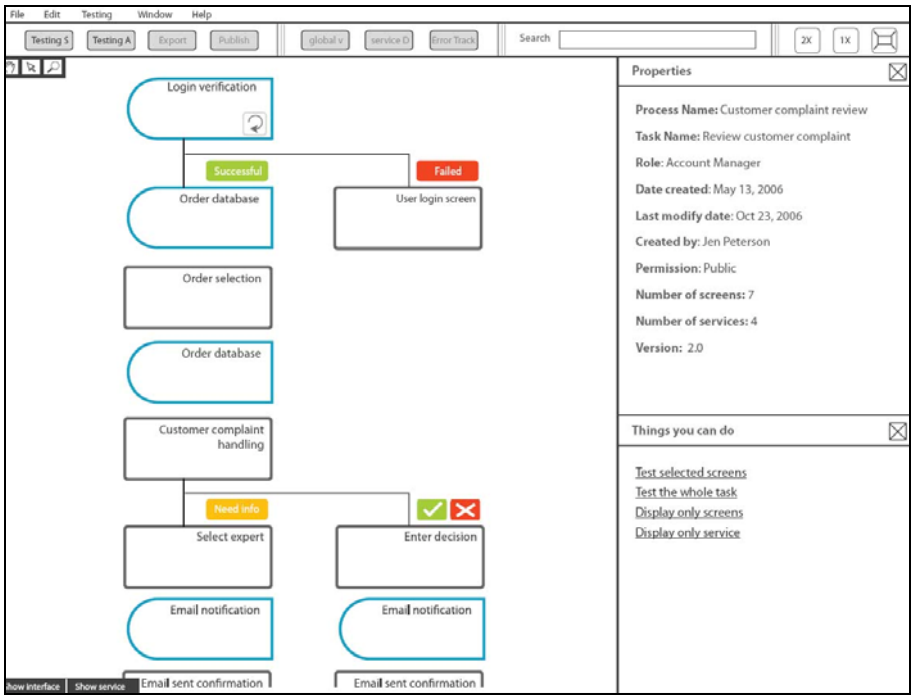


Figure 6-17: Testing environment of P2

Derived from an internal design review session, we recognized that it is important to give users enough space to view the whole workflow to make selections before starting to test. We redesigned so that the workflow canvas was expanded during the time when users were making selection of the path to test before the test session started. Figure 6-17 shows the mock-up of the new testing environment.

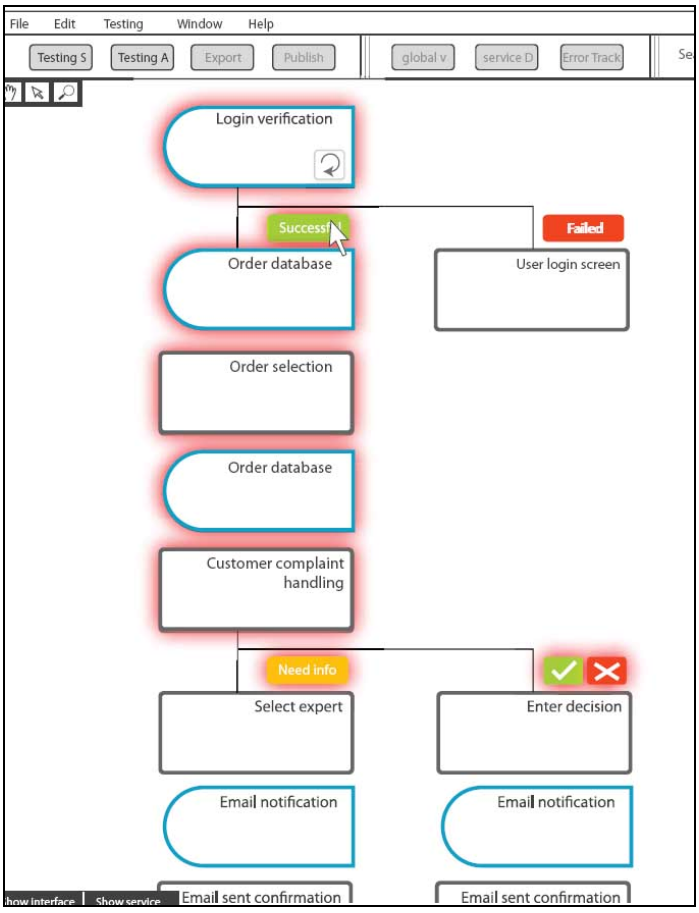


Figure 6-18. Testing environment - workflow selection

Workflow selection method drastically changed in this iteration. The evidence from user research showed that workflows in each task within the process have multiple decision points (see Appendix for detailed user feedback). In this iteration, we allowed users to select different path in the workflow, where a path was composed of many sub-paths. With the support of auto-path-completion, once users selected any component in the sub-path between two decision points, the whole sub-path will be automatically selected. Figure 6-18 show a screenshot of the new workflow selection method with selected path highlighted in red and decision point presented. We also decided to remove the numbering system, since it was proven to be confusing. The removal of these numbers also prevented human errors in entering the numbers.

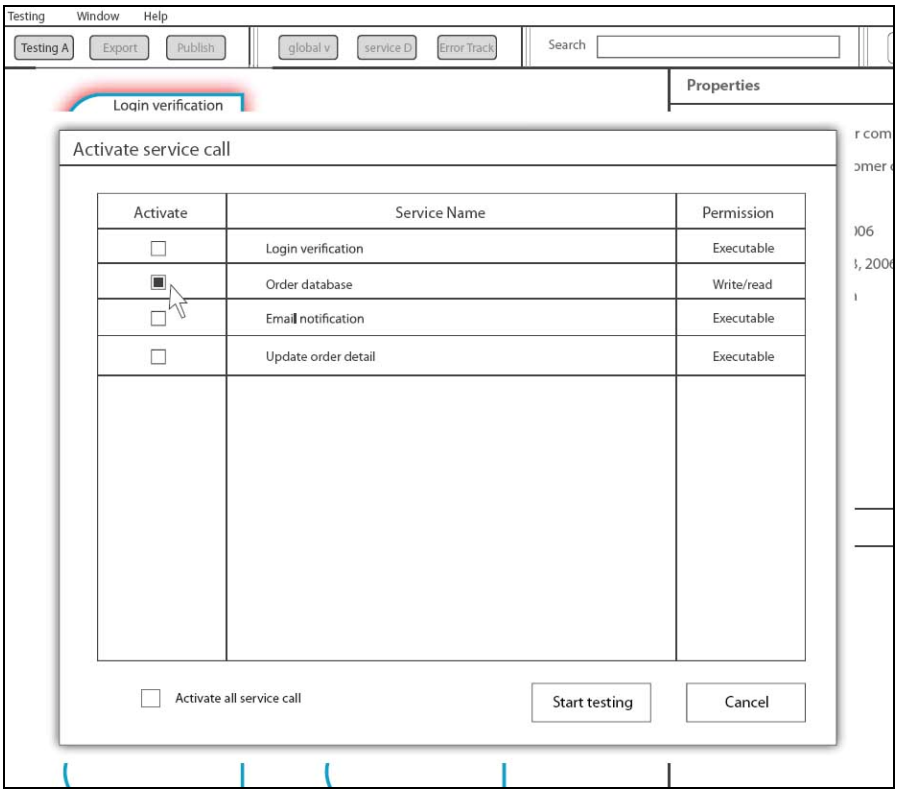


Figure 6-19. Testing environment - service activation

In this iteration, we also adapted the users' suggestions from user testing and allowed users to decide which services to activate upfront. We added a dialog box popup right when users pressed the [Start Testing] button. The dialog box displayed all the required service calls in this workflow, where each is presented next to a checkbox. Users could then select which services they want to call before start testing. Figure 6-19 shows a screenshot of the service activation screen.

Another new feature added in this iteration was letting users set environment variables based on the users' feedback from previous iteration. Once users started testing, the option to set the date and year of the testing environment option would be displayed in the form of a popup box.

The service dependency view designed in previous iteration showed the data flow dependencies to help users gain a better understanding of the process. During user testing, users were excited about this feature, but still felt the information we provide could be more helpful. In this iteration, we improved the service dependency view to show data mapping as suggested by the users. Figure 6-20 shows the screenshot of the redesigned service dependency feature displaying exactly how the input and output data is mapped in the current service or screen.

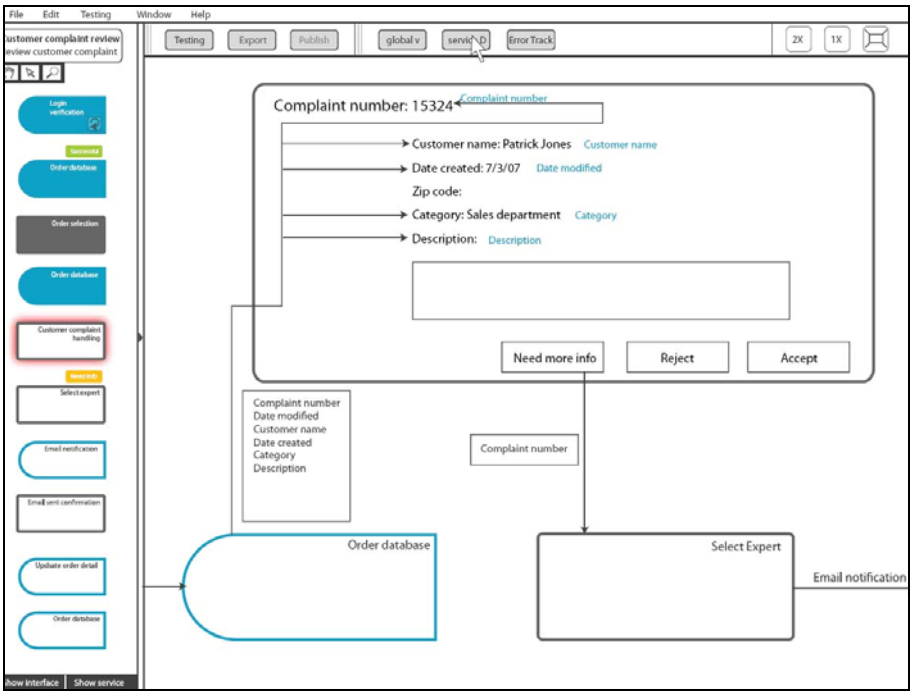


Figure 6-20. Testing environment - service dependency

Another major change was the decision to display service information as an individual screen during testing (Figure 6-21). Since the initial purpose of testing was to imitate the interaction with the actual composite application, we assumed that users should not see the service information. However, several user testing participants conveyed the need of viewing service information; users like the idea of simulating the actual end-user experience, but they also want more information for debugging purpose (see Appendix E for detailed user feedback). We also enabled free examination of any screen and service at anytime as users requested.

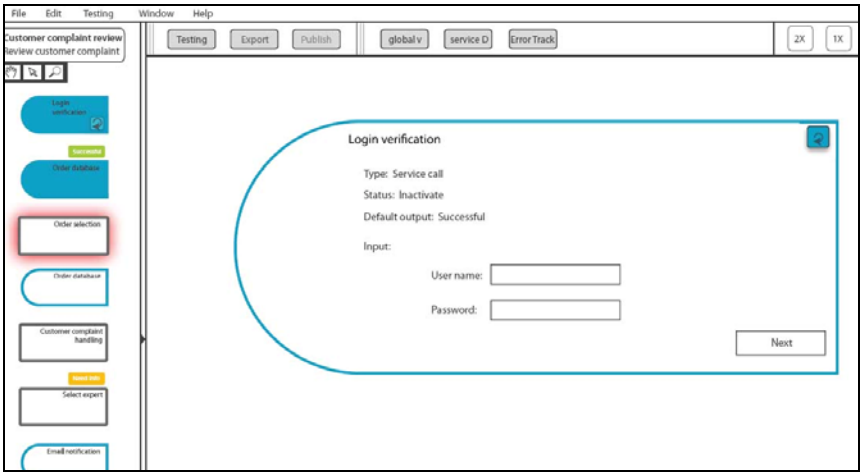


Figure 6-21. Testing environment - service information screen

Since many users were not impressed about the demo video studio concept, we decided to drop this feature (see Appendix E for detailed user feedback). From prior user research, the demand for easy tutorial creation is still high (see Appendix A for details). Based on iteration 1 testing, users suggested that being able to export screens of tested workflows would be very beneficial (see Appendix E for detailed user feedback); and it was evident that people found it a hassle to use print screen

function to capture the screen shots. In this iteration, we incorporated this user feedback and allowed users to export all the screens as screenshots into PDF, which was a commonly used documentation format based on our observation from user research.

6.7.2 User Testing on Paper Prototypes

Similar to the user testing we have done for previous iteration, we performed Think Aloud Studies with four (4) business process experts to evaluate P2. Three of the participants were involved in the user testing of P1, while the remaining one has only participated in our concept validation session.

At the beginning of the study, users were asked to express their own understanding of the differences in process flow versus workflow. Users then were introduced to the workflow visualization in our prototype and asked to perform a think aloud walkthrough of visualizations. Users were instructed to narrate as much information as they can get from the visualization. After this, users were given a broad overview of the system. Finally, users were asked to complete several tasks framed as use case scenarios and articulate their thought processes as they went through the tasks.

Although there were concerns about that a ramp-up walkthrough of the system might bias the user testing results, we felt doing so better simulates the real usage situation where users are expected to get some conceptual training before start using an expert tool.

Our main test goals for this prototype testing revolved around learning how users reacted to the new visualization paradigm and the data mapping mechanism:

- Are users comfortable with the differentiation we imposed on process flow and workflow?
- Do users recognize all the possible things they can do with the Action Library?
- Do users find having a central data registry helpful in constructing data mapping?
- Do users find the interface supports all the functions they need to model a workflow?
- Do the service dependencies provide enough information?
- Does workflow selection make sense to the users?
- Do users find the export screenshots function useful? What else they would like to see on the exported screenshots?
- What other service information is needed?
- Does the service activation dialog box support the users' needs?

The complete collection of direct verbal feedback from the user testing sessions of P2 can be found in Appendix F.

6.7.3 Takeaways

The most important observations we learned from user testing are summarized below.

6.7.3.1. Workflow visualization

Overall, the new workflow visualization was well received. We observed no significant barrier in making the distinction between the workflow and the high-level process. The users seem to understand the workflow as a list of instructions that will be taken in order:

"I almost always prefer to have the workflow goes downward like this"

However, there were still difficulties in understanding which action was a human step and which was a system step.

The users also reacted positively to the rigid workflow construction as opposed to freestyle diagramming, where significant amount of effort is required to maintain a clean and logical layout of the elements in the diagram throughout the canvas space. When asked about whether the visualization would be able to scale well, one of the users validated our assumption about the complexity of a workflow.

"[A workflow] usually would not contain more than 10 to 15 steps. A workflow that has more than that number of steps is probably poorly designed anyway. It will not be maintainable."

6.7.3.2. Action library

All users we observed were able to quickly figure out how to use the Action Library effectively. One of the users pointed out that there may be various versions of the same service listed and the details display should include the creator and last modified date.

6.7.3.3. Defining data mapping

When asked to define the data mapping, users felt the current way of having to go back and forth between the workflow and the Data Library to get the context was somehow awkward. It was desirable, however, to be able to see a list of potential candidates. One of the users suggested that it would still be nice to see a visual representation of all the data flow in the workflow workspace particularly when examining the data dependency.

6.7.3.4. Navigation control

All the participants immediately recognized the Google Map controls at the process level modeling and were able to use the controls to navigate the canvas effectively. However, rather than zoom into the task to see more details, the users would either right-click on the process element and expect to see the "drill-down" option in the context menu or double-click to edit the contents in the process element.

6.7.3.5. Information for exported screen shots

We received positive feedback on the exporting screen shots feature. User further suggested that adding contextual information such as input parameters and screen name for each screenshots would be helpful. We felt that such function can be optional to be turned on or off according to user preference.

6.7.3.6. Test the whole process

Our design enabled users to test each task workflow individually, but not multiple tasks at once nor the entire process. Users gave solid suggestion that being able to test the whole process is extremely important, as well as being able to select multiple tasks and test them at once. We would like to extend the scope of the testing environment if possible.

6.8 High-Fidelity Prototype: Iteration 3 (P3)

After two rounds of paper prototyping, we gained an in-depth understanding of user needs. We then moved on to create the interactive prototype that would eventually demonstrate our final solution. The third prototype (P3) contained most of the essential structure and features that were inherited by the final prototype. Four (4) comprehensive use case scenarios were generated for this iteration. The use cases were made high-level in order to give users the freedom to explore the prototype (see Appendix G for use cases).

6.8.1 Prototype Overview and Design Rationale

6.8.1.1. Look-and-feel

One of the main motivations of this design iteration was to determine the look-and-feel of our design. We also aimed to improve consistency in terms of the tool bar and its location, navigation tools, color scheme and layout style across the three different environments in response to the feedbacks we received from an internal UI review. The internal UI review was held with a number of human-computer interaction specialists aiming to help improving the usability of our user interface design.

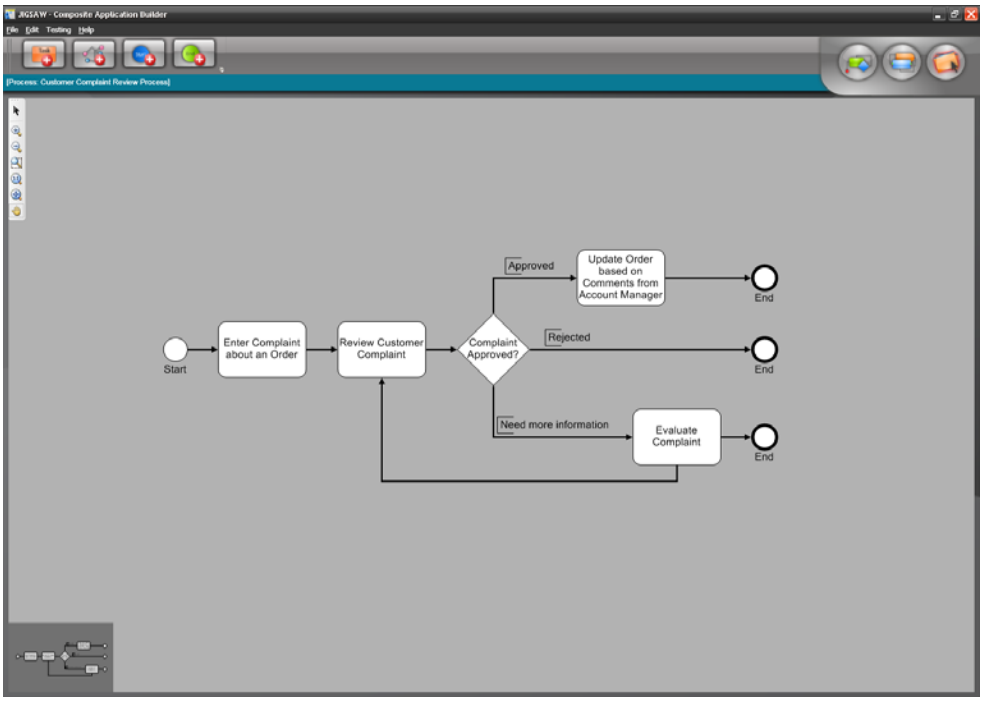


Figure 6-22: The process modeling environment of P3 with newly designed toolbar and look-and-feel.

The concept of two different types of toolbars (the navigation toolbar and the system toolbar) emerged in this iteration and was later adapted in the final design. Figure 6-22 shows the initial design of the toolbars in the process modeling environment.

We also spent time on picking out appropriate color schemes for our system. Human perception theory informed us that dark backgrounds do not disturb the user from focusing on the main task. Thus, we decided to use a dark color scheme as shown in Figure 6-22.

6.8.1.2. Workflow Modeling Environment

After experimenting with two different styles of workflow visualization in the previous two iterations, we decided to use the visualization from iteration 1. To help users further differentiate elements, we also used a different color for each different connection link and element (Figure 6-23). The workflow modeling visualization from iteration 2 did not convey enough information visually; based on the feedback from the UI review session, the visualization from iteration 2 was evaluated to be hard for user to differentiate between screen flow and data flow when they were pressed together. Moreover, the visualization of iteration 2 did not appear to have the potential to support complex workflows, which was raised as a concern during the UI review session.

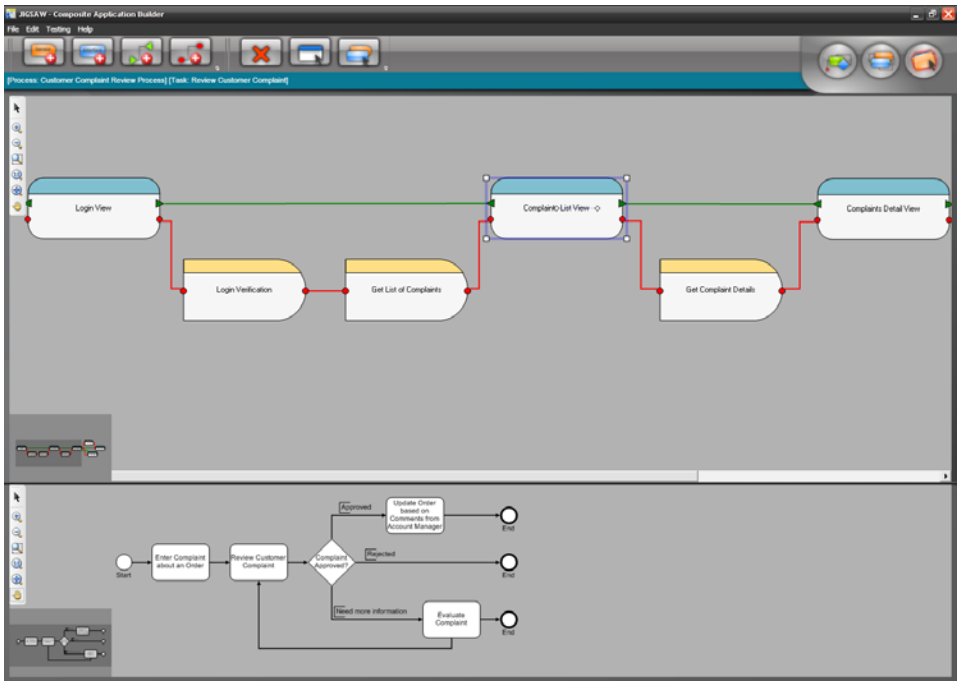


Figure 6-23: Workflow modeling environment of P3 with horizontal visualization that was similar to P1.

The concept of the Action Library from iteration 2 was adapted as the Service Library and the Screen Library. We simplified the categories into two to clearly correspond to the visualization. We also kept the libraries hidden until users intentionally open it to preserve maximum screen estate for workflow editing. Other concepts such as the Data Library and next-step-suggestion were cut back mostly due to scope constraint.

After experimenting with several different mechanisms of switching between the three different environments, we decided to adapt the Microsoft Office 2007 round button style: three buttons which represented Process View, Workflow View and Testing View (Figure 6-24). These three buttons were placed on the top-right corner for the location introduced the least amount of interference with other tools in the interface.

Figure 6-24: The three buttons for switching between the Process View, the Workflow View, and the Testing View in P3.

6.8.1.3. Testing Environment

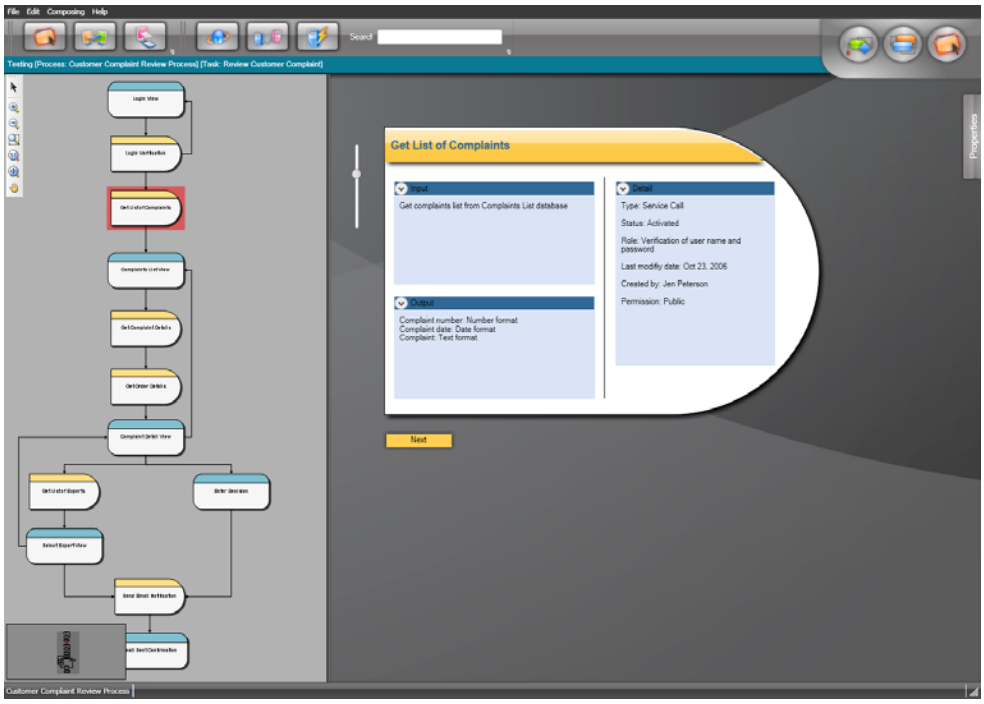


Figure 6-25: Testing environment with new look-and-feel in P3.

We also applied the similar look-and-feel to the testing environment as shown in Figure 6-25.

6.8.2 User Testing on Interactive Prototypes

We performed Think Aloud studies with three business process experts to evaluate the interactive prototype. Two of the participants had participated in the user testing of iteration 1, while one participant had only participated in the concept validation session.

At the beginning, we probed the users about their understanding of major concepts such as the difference between processes, tasks, and workflows. Luckily, the users expressed the same mental model we used in our high-fidelity designs. Each user was required to accomplish four tasks. They were encouraged to play and freely interact with the prototype. We asked them to speak aloud while interacting with the prototype and gave them an opportunity after the test to ask questions.

A benefit of having an interactive prototype is that it allows the observation of mouse pointer movements correlating user feedback. We learned from these observations that users responded positively when encountering animation effects

such as button highlighting on mouse over gestures or the animation of information panels.

The complete collection of direct verbal feedback from the user testing sessions of P3 can be found in Appendix G.

6.8.3 Takeaways

The most important observations we learned from user testing are summarized below. These takeaways informed the changes to the design and helped shape our final design solution.

- The users pointed out that even when it was intended to build composite application to support the business processes, there might be some manual tasks that would not be captured in a composite application. Hence, the process modeling should allow the modeling of tasks that do not involve interaction with composite application to maintain the completeness of a business process model.
- The workflow canvas in the testing environment remained vertical, while the workflow in the workflow modeling environment was horizontal. It was important that we keep them consistent as one style.
- We experience some challenges in designing icons that would resonate the users' mental representation of the intended usage. Particularly, the users had a hard time to associate the [Start Testing] button icon with its functionalities.
- The navigation overview map was proven to be effective. Although the users were not able to identify the usage of the overview map at the first glance, they quickly picked up the functionalities as soon as they started interact with it. The users expressed that they prefer using the overview map to pan around the canvas than using scrollbars.

7 Conclusions and Future Work

The design we created is based on user data and iterative analysis. Our design allows user to first design the high-level process and then drill down to create workflows for individual process steps. The key advantage of our design is that it allows users to visually compose and configure their composite application, and this makes it very appealing for users with minimal IT know how.

While we made great strides in our design, there are several design ideas that were driven by initial research that could not be incorporated into our prototype due to time constraints. We believe that these ideas would greatly improve the value of our prototype.

- **Documentation Generation**

  Although we have provided users with an intuitive way of designing their composite applications, business process experts with little IT know how would like to receive guidance on specifying implementation details that are missing in their design. This may be accomplished through a wizard that guides users to fill in all required information or using contextual alerts that notify users when certain details are missing.

- **Documentation Management**

  With the increasing pain of document management, users greatly appreciated the idea of attaching documents to corresponding process steps. They also wished to be able to add annotations and comments to the process steps to specify requirements and design rationale. This would make it quicker to find, access and retrieve documents. Additionally, the system should allow users to set access control restrictions to documents.

- **Best Practice Communities**

  During our user research phase, we explored the use of best practices during business process modeling, and found that users often used best practices to ensure that their design was efficient. However, it was important for them to validate the credentials of the creator before they would customize it for their purpose. This can be achieved by an open non anonymous forum incorporated within Jigsaw where users could post their solutions and ask questions. An optional way to achieve this would be to provide contextual suggestions of best practice to users modeling a business process.

- **Tracking Project Initiatives**

  Adding functionality to our system that helped users track business process implementation was out of our project scope, but we received feedback from users that this functionality would be very advantageous. Multiple users we spoke to were in the process of implementing this functionality in their organizations, and incorporating this feature in Jigsaw could prove valuable.

- **Adding to Libraries**

  Currently, our solution only addresses reusing existing services and screens to create a composite application. However, a possible future direction could be to allow users of Jigsaw to connect to an external system from within Jigsaw, to add new services and screens. This could include allowing users to

take advantage of the UDDI (Universal Description, Discovery and Integration) protocol to find and use web services over the Internet.

- **Flexible Business Processes**

  We found that while high-level process remain static, details of business processes change rapidly and often those practicing the process, do not have the authority to change them according to their needs. Thus, it would be important to empower these end-users to violate business process rules. One can imagine that over time with multiple recurrences of a violation, the process adapts to include the violation in its flow.

- **Other Potential Users**

  We found that the attributes of the business process expert that led to the design solution also apply to other potential users such as an end-user. These attributes include the non-technical nature of business process experts and their workflow centricity. Thus, it can be hypothesized that Jigsaw can also be used by these end-users. In the future, we would strongly suggest research of end-user needs to make Jigsaw available to them.

References

[1] SAP tops CRM and ERP market, Steve Ranger, http://software.silicon.com/applications/0,39024653,39156716,00.htm, Accessed March 7, 2007.

[2] Business Process Cycle Overview. https://www.sdn.sap.com/irj/sdn/bpx-cycle.

[3] Andrew J. Ko, Brad A. Myers, Htet Htet Aung, "Six Learning Barriers in End-User Programming Systems," *vlhcc,* pp. 199-206, 2004 IEEE Symposium on Visual Languages - Human Centric Computing (VLHCC'04), 2004

[4] Apple – Mac OS X – Automator. http://www.apple.com/macosx/features/automator/.

[5] JavaBeans Specification, Sun Micro Systems Inc., http://www.ibm.com/developerworks/java/library/j-what-are-ejbs/part1/, last accessed 07-29-2007

[6] Object Management Group, Business Process Modeling Notation (BPMN), http://www.bpmn.org/, last accessed 07-29-2007

[7] Microsoft Expression Tools, http://www.microsoft.com/expression/, last accessed 07-29-2007

[8] Extreme Programming Software Development Methodology, http://en.wikipedia.org/wiki/Extreme_Programming, last accessed 07-29-2007.

[9] Program library ILOG Diagrammer for .NET 1.0, http://www.ilog.com/products/diagrammernet/, last accessed 07-29-2007.

[10] Liu, N., Hosking, J., and Grundy, J. 2004. Integrating a Zoomable User Interfaces Concept into a Visual Language Meta-Tool Environment. In Proceedings of the 2004 IEEE Symposium on Visual Languages - Human Centric Computing (Vlhcc'04) - Volume 00 (September 26 - 29, 2004). VLHCC. IEEE Computer Society, Washington, DC, 38-40. DOI= http://dx.doi.org/10.1109/VLHCC.2004.32

[11] Frye, Colleen. The role of composite applications in an SOA. July 27, 2005. SearchSAP.com. http://searchsap.techtarget.com/originalContent/0,289142,sid21_gci1111132,00.html

[12] Extensible Application Markup Language, Wikipedia Entry, http://en.wikipedia.org/wiki/XAML, last accessed, 07-29-2007.

[13] Brad A. Myers, Scripting Graphical Applications by Demonstration, Proceedings of the SIGCHI conference, 1998, http://portal.acm.org/ft_gateway.cfm?id=274716&type=pdf&coll=GUIDE&dl=ACM&CFID=26386115&CFTOKEN=65280711, last accessed 2007-08-08.

[14] Walkthrough: Hosting a Windows Forms Composite Control in Windows Presentation Foundation, http://msdn2.microsoft.com/en-us/library/ms750944.aspx, last accessed 08-03-2007.

[15] MSDN Forum thread reporting a bug in the drag and drop interoperability between WPF and WinForms, http://forums.microsoft.com/MSDN/ShowPost.aspx?PostID=311581&SiteID=1, last accessed 2007-08-08

Appendices

Appendix A: Contextual Design Models

Sequence Model

Flow Model

Cultural Model

Artifact Model

Appendix B: Final Design Use Cases

Requirement Matrix

Use Cases

Appendix C: Concept Validation

Concept Storyboards

Validation Session Feedback

Appendix D: Wireframes (P0)

Appendix E: Lo-Fi Prototype 1st Iteration (P1)

Use Cases

Paper Prototype

User Testing Results

Appendix F: Lo-Fi Prototype 2st Iteration (P2)

Use Cases

Paper Prototype

User Testing Results

Appendix G: Hi-Fi Prototype (P3)

Interactive Prototype Screenshots

User Testing Results