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

Index

1 Executive Summary

As one of the leading business application and Enterprise Resource Planning (ERP) solution vendors, SAP AG introduced SAP NetWeaver, a new technology platform that enables service-oriented composite application and services integration. The SAP NetWeaver platform is designed to integrate business processes by composing web services within the existing IT infrastructure to create new applications, crossing the boundaries of the isolated legacy systems in place today. Currently, application functionality does not fit with business processes. As a result, users are forced to adapt their work practices to the software instead of the other way around. Thus, a gap is created between how business users work and what development organizations deliver. 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 team of Master's students in the Human-Computer Interaction 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 over a period of 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, literature review of various academic and scientific papers, study of the competing products, user interviews and an extensive design ideation process. During Phase 2, we developed design concepts for user evaluation and iterative 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.

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 satisfy identified user needs, the team brainstormed design ideas to illustrate 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, we analyzed user feedback and decided to focus on the concepts of multi granularity views, simulation and documentation management. The team created three rounds of wireframes that detailed these concepts and portrayed various use cases. After each round, we aggregated and analyzed the data, 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, due to the complexity of business processes, users often wanted to know where they were in the overall process. To support this need, our system provides an overview of the process at all times.

After thoroughly testing our iterative 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 were quite successful in designing Jigsaw that is well grounded in user intents and 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. They had their biggest success with the rather 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 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.

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 often situate 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 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.

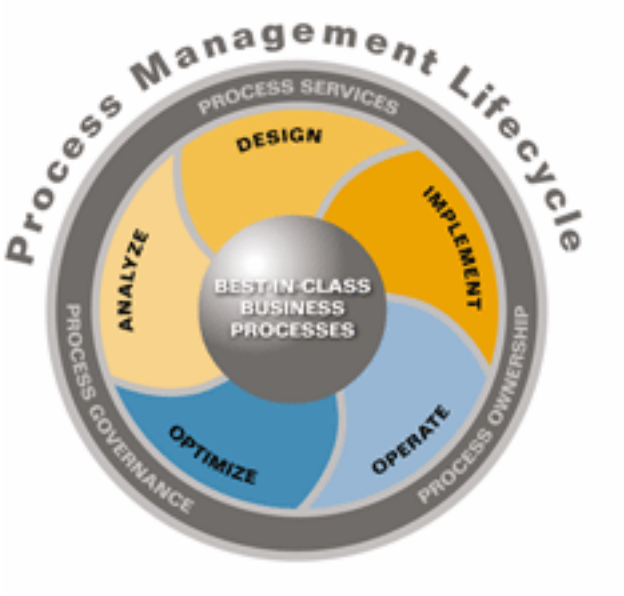

*Figure 3-1 Depiction of Business Process Lifecycle [**2**]*

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 detail steps 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 different 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 Design Process, Setting Design Foci) 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 between these two levels of magnification in the work practices of business process modelers. In particular, SAP tool users often have to create the visual 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 the business process experts and the developers (see Consolidated Flow Model in Appendix A).

Our user research revealed that this problem has its origin in the diverged needs from 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 diagram 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 is highly utilized by the business process experts to visualize the responsibility hand-off between different roles and to spot bottleneck 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 paradigm provided in the commonly used tools, such as Visio and Google Maps, might greatly reduce the learning curve on 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 understand 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. 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 emitted by other 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 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 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 considered to allow users to quickly generate a prototype 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 suggestion of 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 Design Process and Rationale

In this section, we walk through all the intermediate steps that lead us to our final design and prototype. For each step, we discuss the methods we applied, findings that refined our solution, challenges we encountered and our leanings. Complete wireframes for each iteration along with user feedback can be found in Appendix E, Appendix F and Appendix G.

4.1 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 to each individual idea. In order to keep the creative level high, at this phase 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.

4.2 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.

4.2.1 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 4-1).

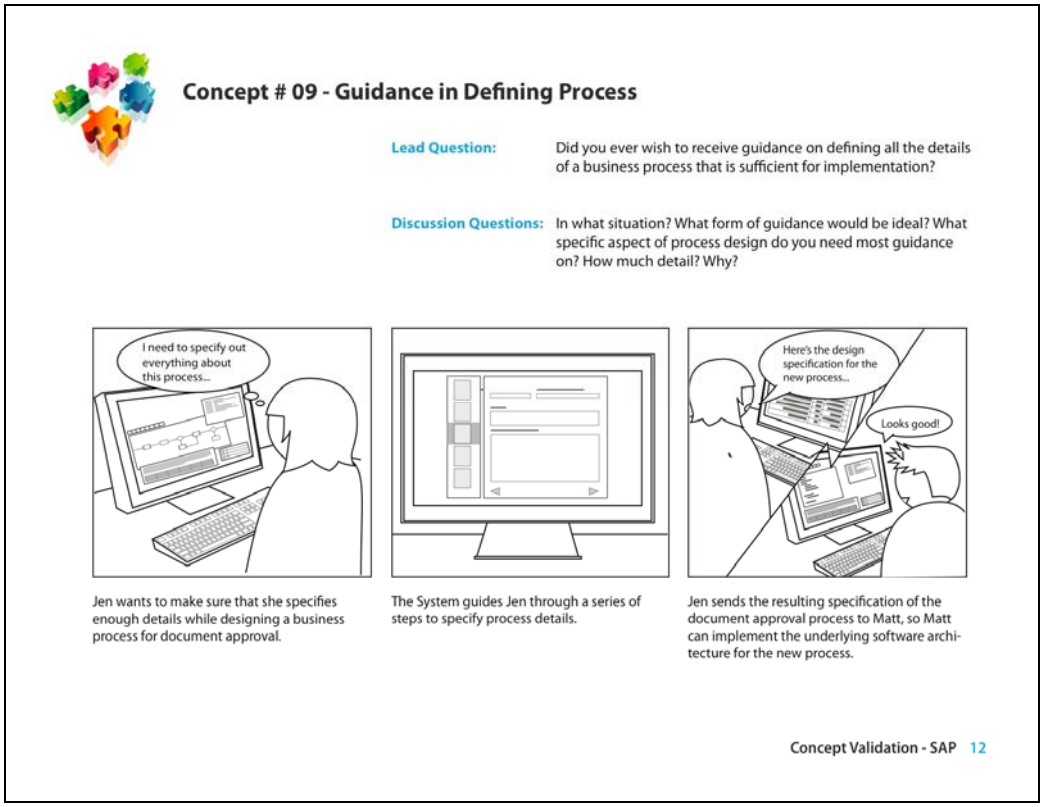

Figure 4-1 A concept validation storyboard contains a short scenario, lead question, and discussion questions. The storyboard is used to find the overlap between the needs researcher observed and needs user percieved.

4.2.2 Concepts

These storyboards covered a rich variety of our design ideas inspired by the design implications derived during the research phase, which 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

The full set of concept storyboards can be found in Appendix C.

4.2.3 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.

In result, most of the concepts were well received. The collection of direct feedback from users can be found in the Appendix C.

4.3 Setting Design Foci

After our concept validation session, the team was faced 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 in 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 get modeled with concrete specifications and is evaluated against the business users needs. We visualized 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.

4.3.1 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.

4.4 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 ground in discussing how all the different features can be pulled into one system and act together coherently.

4.4.1 Features

Below listed 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 freestyle diagramming paradigm
- Reusing 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 controls
- Visualizing the dependency between different processes over a process flow diagram
- Attaching annotating documentation to a process flow diagram with color coding schema.
- Composing a workflow demo in video format

4.5 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 for us to further prune down the scope of our design.

4.5.1 Feasibility-Novelty Matrix

Our team leveraged the method of cost-value matrix to help evaluate the features. In order to tailor to our project goal, 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 sticky notes 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 4-2 shows the resulting matrix. Blue sticky notes are individual placement; yellow sticky notes are consolidated placement.

Figure 4-2 The feasibility-novelty matrix. Blue sticky notes are individual placement; yellow sticky notes are consolidated placement.

4.5.2 Takeaways

After the initial round of wireframes, we came to the following three design foci:

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

4.6 Lo-Fi Prototype: Iteration 1 (P1)

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

4.6.1 Prototype and Design Rationale

4.6.1.1. Composite Application Composer

Our first prototype (P1) tested the idea of modeling composite application using two-level flow diagram visualization. At the top level, a user could model high-level business processes in a way that was similar to the one in Visio or ARIS; at the second level, a user could compose the workflow in a way that was similar to the one in Visual Composer. This idea is derived from the design implication of leveraging flowcharts as the visualization of workflow. Figure 4-3 illustrates the notations we used.

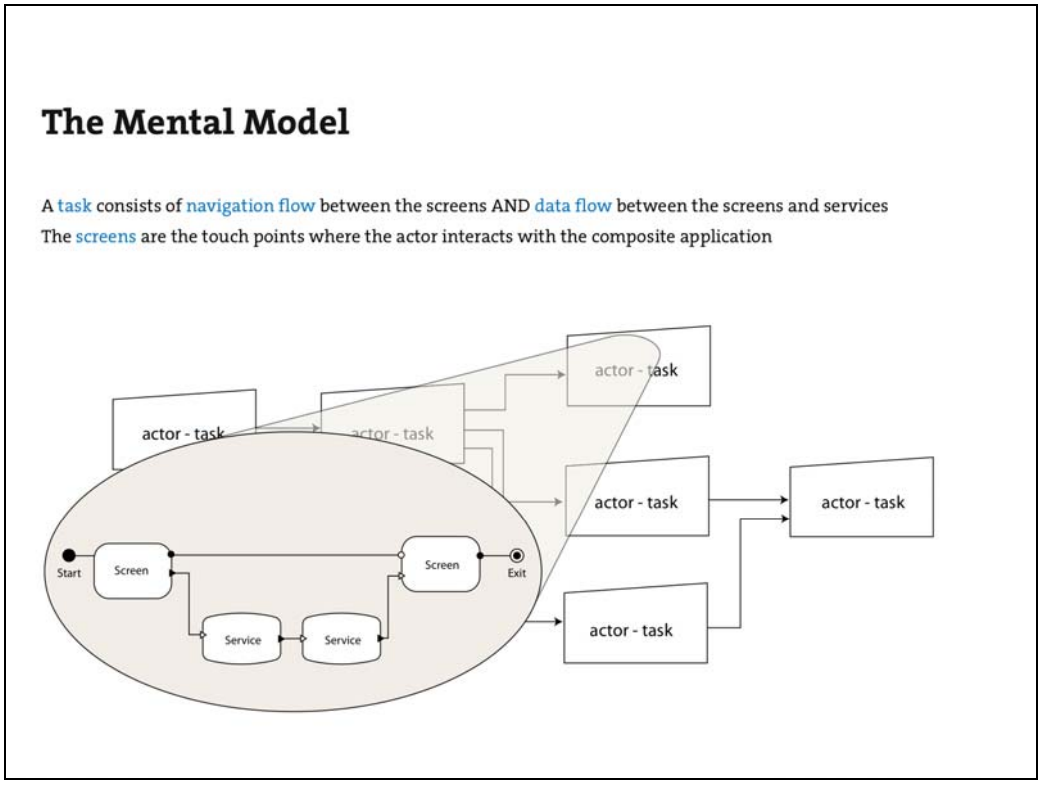

Figure 4-3 We simplified and customized a set of notations to form a visual representation of the composite application in P1.

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. In our system, a process was merely a sequence of tasks, a high-level step that was taken by one actor independently.

At the workflow level inside a task, there existed two types of flow diagram elements, screen and service. The screen represented the human step in the workflow, whereas the service represented a system step. The relationship between those elements is established by two types of connections: navigation line and data flow line. We leveraged most of the workflow-modeling paradigms from Visual Composer in this version of prototype, since Visual Composer was originally designed to target business analysts. 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 next level.

We have observed from user research that business process experts often first model the process at a higher level of abstraction, and then elaborate on the details. We believed that the clear separation of functionality at different levels would support users' working pattern. The limited number of functionality at each level also will ease the mental workload on making decisions of what to work on.

The multi-level workspace also naturally incorporates the usage of pre-built templates (Figure 4-4). In this prototype, the system presented users the choice of templates available whenever the user wants to create a new workflow or a new screen. It is evident that people never start designing a new process or application or user interface from scratch. People often based their design on existing works that serve similar function as the basis of modification and improvement. A number of

end-user programming studies also indicated that seeing examples helps users making design decision.

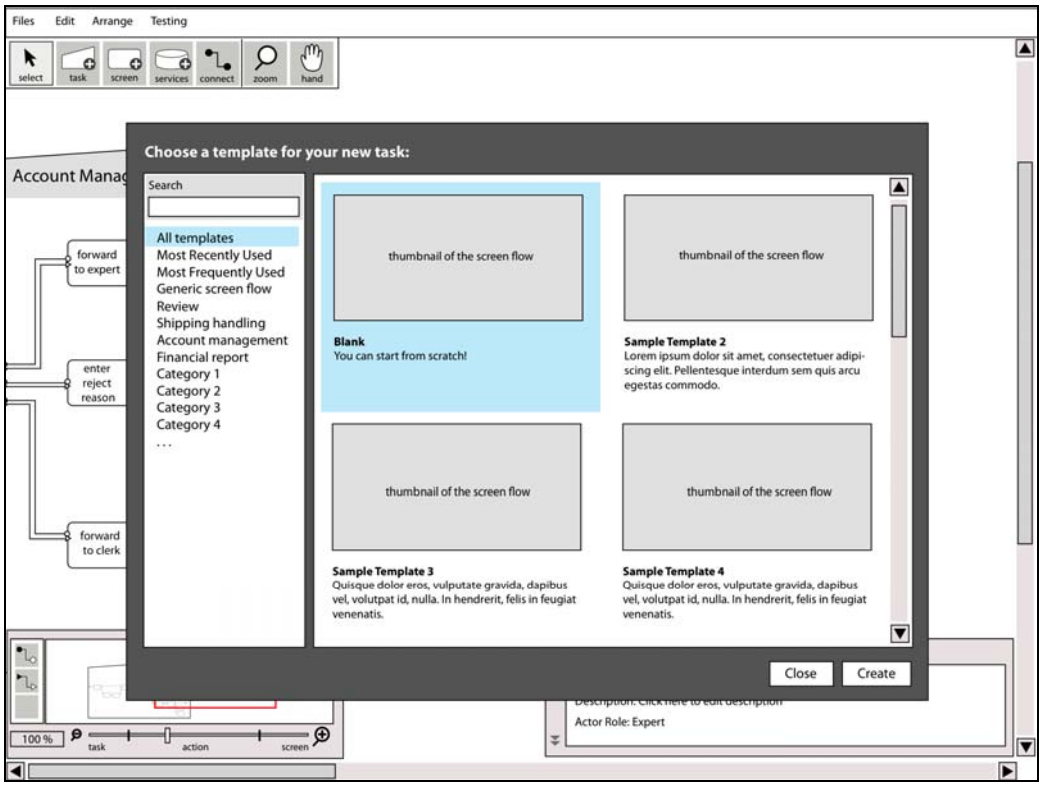

Figure 4-4. The template chooser allows users to browse or search through a huge library of pre-built best practice templates as the basis of a new workflow model.

The concept of zoomable user interface (ZUI) and multi-granularity view of the process is 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 (or de-zoom) 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 4-5 shows the zooming controls we proposed in this prototype.

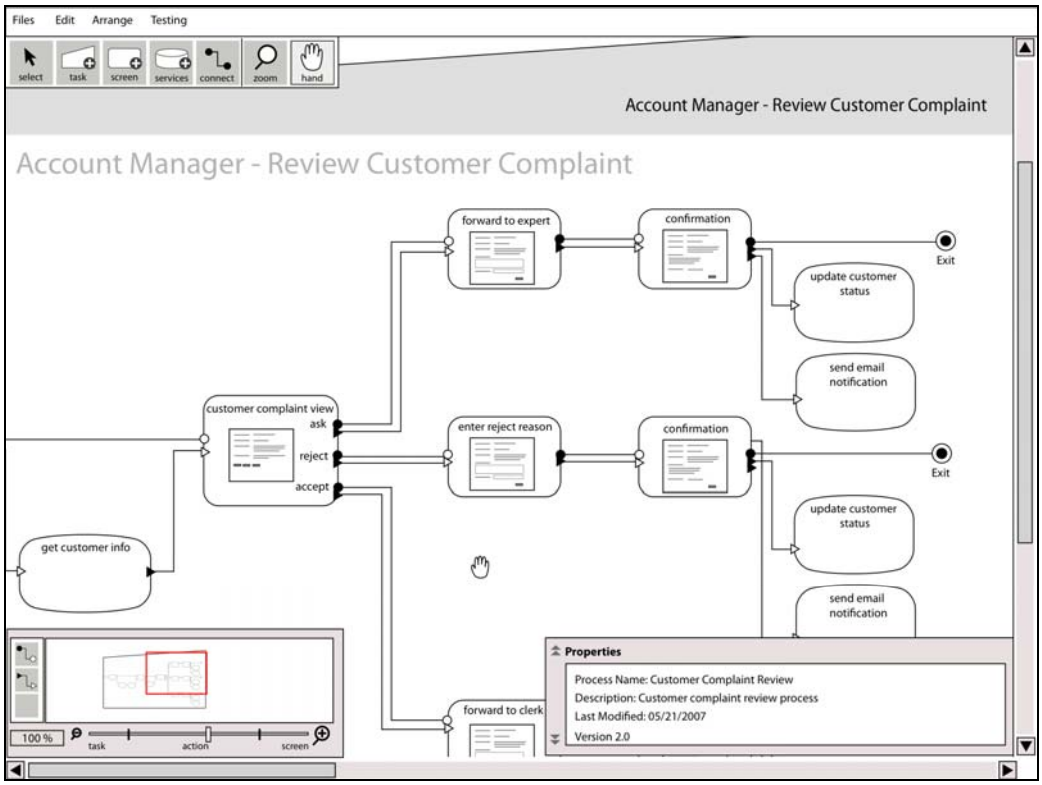

Figure 4-5. In P1, users can use the slider at the lower left corner of the canvas to zoom-in or zoom-out. There are also navigation tools in the toolbar at the top of the canvas.

4.6.1.2. Testing Environment

Another component of our prototype introduces the concept of business process simulation, which outputs an interactive prototype of the business process. This prototype acts as a decision-making tool, which imitates how the process would work after it has been deployed. It helps the business process designer to gain a better understanding of both the process and end user behavior at multiple levels. On the interaction level, the business process designer can enter data input just as they would in real applications. On the conceptual level, our tool provides a detailed view of service data mapping between multiple screens and services to represent the data flow. This concept originates from the needs of testing business processes before deploying them into production systems. During our user study, multiple business process designers have expressed the importance and the pain of process testing, and the common wish for test automation (see Appendix E). Figure 4-6 shows the screenshot of the simulation environment.

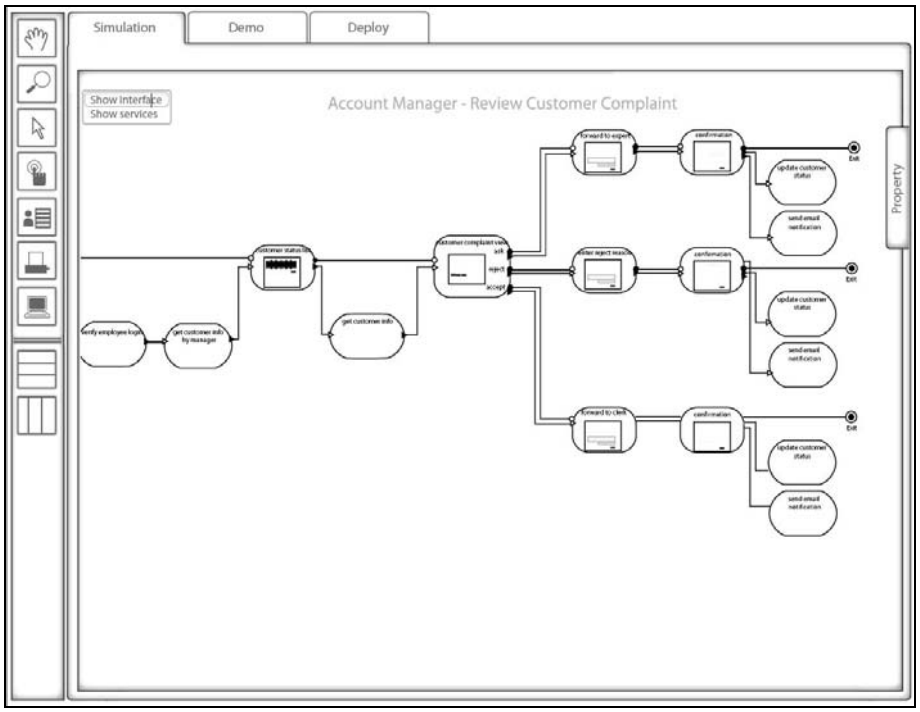

Figure 4-6: Simulation Environment

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. In this prototype, we provide both styles of testing. The system offers the user freedom to choose which the starting point of the simulation, but we prohibited free form selection; instead it provides auto-selection mechanism, which auto selects related screens, services or tasks based on the start selection. This helps prevent errors caused by disconnected data input and output from services and screens. We also proposed a number system for the workflow, by which each screen and service will be assigned a unique number, which the user can enter in the simulation text box. Figure 4-7 shows the number system in workflow selection.

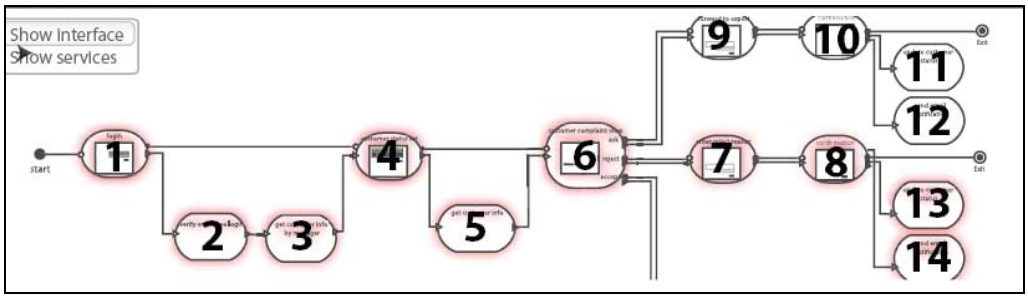

Figure 4-7: Simulation environment - numbering system

The ability to detect flaws in the process is also incorporated in our system. A flaw here implies workflow mis-connection, dangling tasks, screens or services. Based on our study, once the complexity of the process or workflow goes up, it is rather difficult to spot these problems (see Appendix E for testing results). Our system highlights these problems in different colors. Moreover, to make the workspace more flexible, once a flaw detected; users can easily switch to composer view to edit the workflow. Figure 4-8 displays a close view of this function.

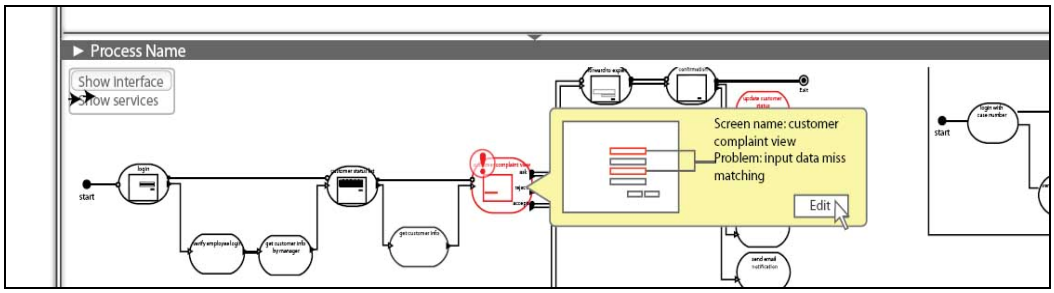

Figure 4-8. Simulation environment - flaw detection

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. Based on an input field parameter, we have proposed that the system alerts users for false input data type. This idea is inspired by the theory of run time error detection. It is evident that it helps user to detect possible data mis-mapping or false data parameter definition. Because most processes involve cross process service call, it is evident that occasionally some essential service call cannot complete due to a broken or inactive service (see Appendix E for details). To help eliminate these issues, which most likely will prevent the user from continuing testing, our system allows user to input customized data for missing services calls. Moreover, our system saves this customized data and passes them along to help the user complete prototype testing. Figure 4-9 gives a close view of how the function works.

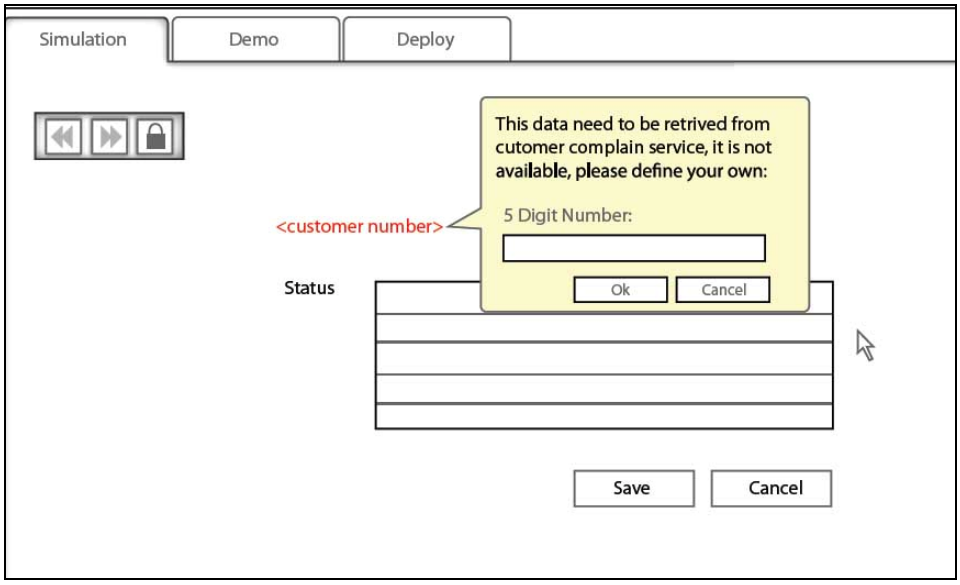

Figure 4-9. Simulation environment - input customized data

Based on the data from user study, business process experts consider that being able to identify where you are in the process is very crucial to business process design. In this prototype, the system provides a navigation view of the entire tested workflow, with the current screen or service you are on will be highlighted. Additionally, business process experts also expressed the need to jump around within the workflow while testing. In our proposed concept, we prohibit the user from jumping ahead in the flow to eliminate the risk of having missing input data. For detailed testing results, please refer to Appendix E.

This prototype also introduces the concept of the service dependency. A user could view both services and screens. Data flow here involves a higher-level view of the where all the input data comes from and outputs. This feature enhances the concept of data flow connection, at the same time, we believe that providing the detailed view of service connection helps users to debug the data flow. Figure 4-10 shows the visualization of service dependency.

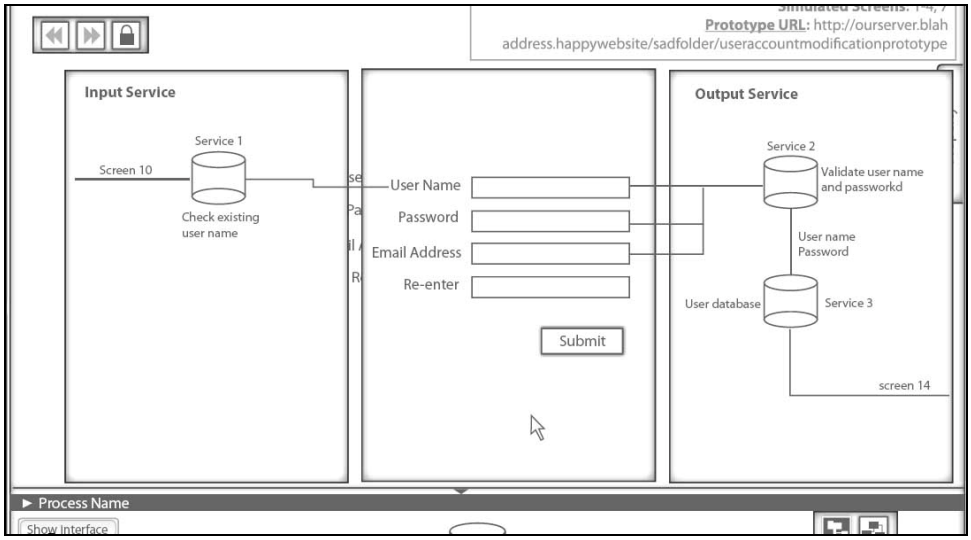

Figure 4-10. Simulation environment - Service dependency

During our user research, we discovered that the new process would always be tested among multiple stakeholders before putting it in production (see Appendix A for details). We suggest the concept that a prototype can be published, in which the system provides a simple URL for business process experts to pass around.

4.6.1.3. Demo Video Studio

The last part of this prototype introduces a video editing tool, which is especially designed to create process demo videos. This demo video studio consists of two parts; the first part enables users to record audio sound and screen movement; the second part provides basic video editing tools, using which a user can modify the recorded files and make it more understandable to end users. Figure 4-11 shows the screenshots of demo video studio.

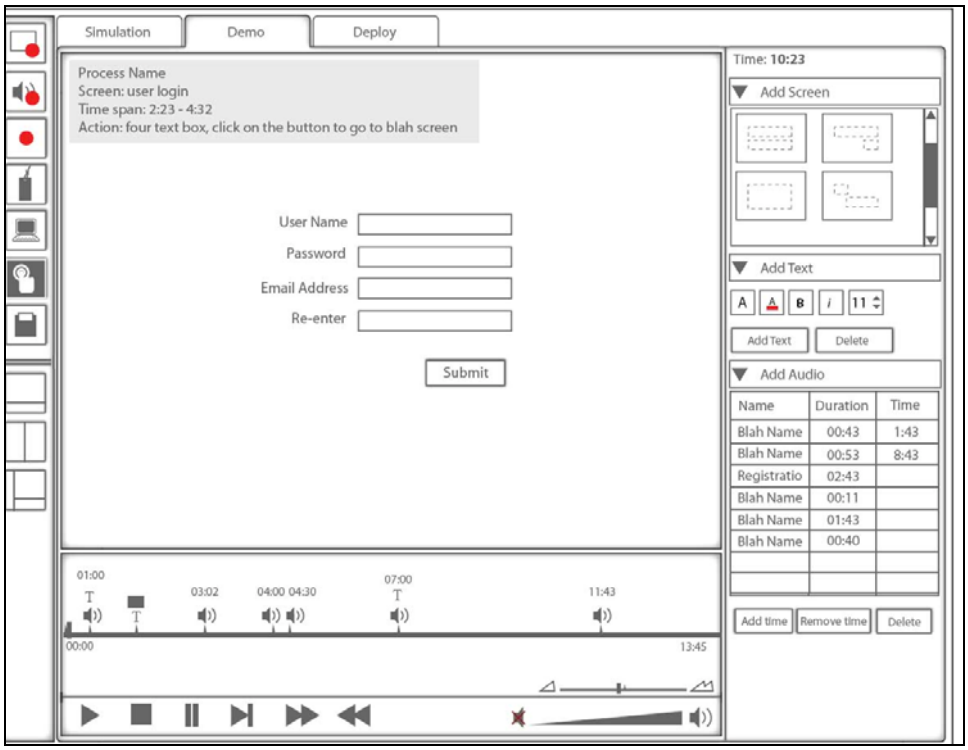

Figure 4-11. Demo video studio

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 prototype, our system combined both audio and screen recording tools and 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 is 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 believe that a simple interface with only the necessary functions can help users save time.

4.6.2 User Testing

We performed Think Aloud Studies with 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 action. 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 user reaction.

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 inquired users about desired functionalities, understandings on the toolset usage, language used, and other relevant feedbacks. The complete collection of direct verbal feedback from the user testing sessions can be found in Appendix E.

4.6.3 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.

4.6.3.1. Flowchart and automated 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 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 rush 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 wanted to seek for alternative visual representation.

4.6.3.2. 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 element and screen element were not distinctive enough to differentiate in a glance.

4.6.3.3. 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.

4.6.3.4. Simulation concept misleading

One of the main difficulties arose from the mismatch between the user's mental model of simulation and ours. Our system proposed that simulation is a step, which produces an interactive prototype. User can share and interact with the prototype. However, users tended to interpret 'simulation' as the interaction with screens'. Moreover, the word simulation itself did not make enough sense based on the functions we proposed.

Furthermore, other users were not familiar with the concept of prototype, in which we introduced that the interaction level of prototype parallels with real web application. To facilitate the usage of our system, we saw a need to change our conceptual model of simulation.

4.6.3.5. Work flow selection

A common response we received was that the workflow model is similar to a decision tree, and users found it hard to perform auto-selection on related screen or service based on the current user selection, since the system could not suggest which path to take at decision points. The current prototype ends up auto-selecting the rest of the workflow after the decision point. User suggested making each path in the workflow into different routes, and once a decision point is reached, the choice of routes is displayed.

4.6.3.6. Service activation

There was a mixed reaction to service call activation in user testing. Some users said that they would prefer to disable service call at all times, and input customized input data. Others prefer activate the service call during all time, to ensure whether the call would be executed successful. Many users value real life imitated interaction, by enabling service call at all times, since by doing so they are able to understand how the process would behave after it is deployed.

Additionally, users raised their concerns about any known broken or inactive services, and whether the system would still perform the service call at this point, or return a failure message. Users suggested having the option to choose which service they want to active.

4.6.3.7. Set up the simulation environment

A few users pointed out that sometimes there is a need to set up the simulation environment, such as when a business process expert is trying to run a simulation based on the data from certain period in the past. It is evident that most commonly

used environment variable in this case would be time. For this matter, user requested to be able to set up the time for a specific process simulation.

4.6.3.8. Display service information

Most of the users expressed a strong need to view service details during simulation. While services usually require input data, if the simulation starts with an inactivated service call, customized input data would be required from users. At this time, users need to be able to enter missing service output.

In more complex workflows, services can contain decision points. We learnt that for the next iteration, we would definitely need to take this under consideration.

4.6.3.9. Export user interface screens

Users in general were not very excited about demo video studio. One user revealed that tools that records screen movement and audio sound already exist, and most people already use those tools. It was evident that users were comfortable in using the tools they use currently and were not willing to change.

Users recommended that rather than making the demo video, there is a potential to allow the export of each user interface screen involved in the tested workflow. The purpose of this would be to use those exported screens in creating a tutorial for new processes.

4.6.3.10. 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, user 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.

4.7 Lo-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 on lo-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 more 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).

4.7.1 Prototype and Design Rationale

4.7.1.1. 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 4-12). A workflow, in the design of this prototype, consists of sequence of actions in the order they would be triggered.

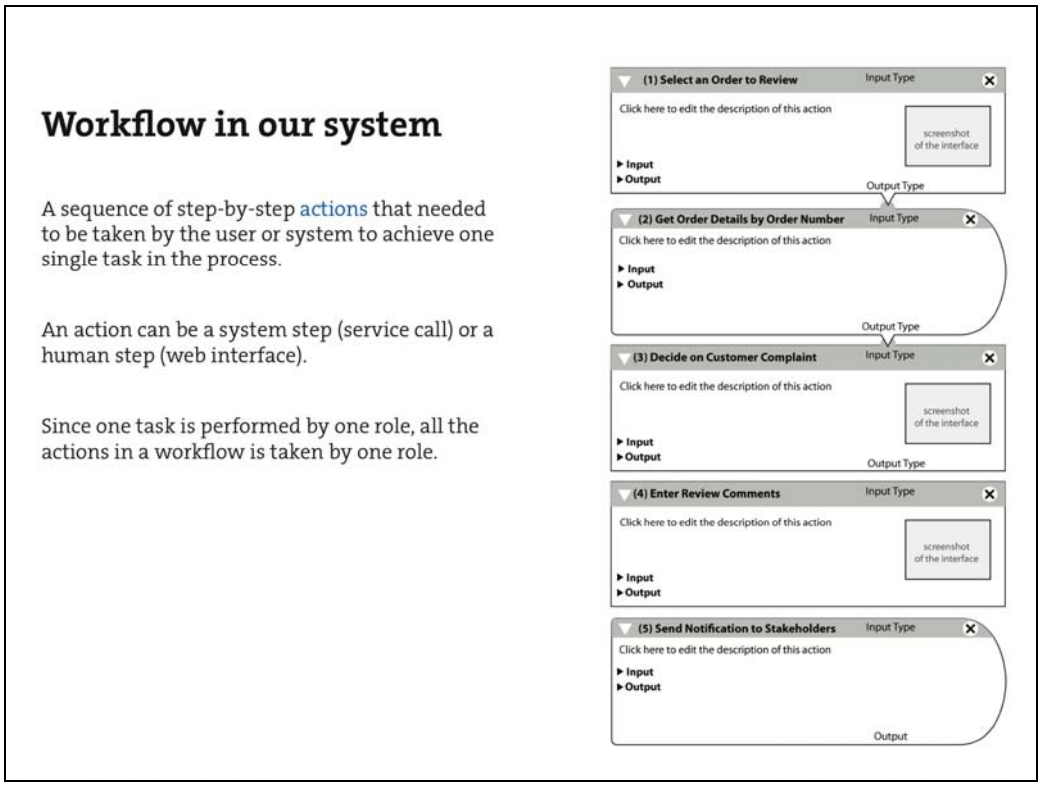

Figure 4-12. This iteration leveraged the Apple Automator visualization of a workflow. The square action element represents a human step (e.g. web user interface, adobe interactive forms, etc.). The action with round side represents a system step (e.g. web service call).

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 renounced the affordance in creating improvident, over-complex workflow structures. It was our intention that users should structure the high-level business process in a way such that each automated workflow model remains independent and maintainable as dictated best practice for composite application design.

Adding a new step to the workflow in iteration 2 also changed dramatically in comparison to iteration 1. With the Action Library (Figure 4-13, see Appendix F), all the possible workflow augmentations were unified into one single series of simple interactions, namely search, select, and then drag-n-drop.

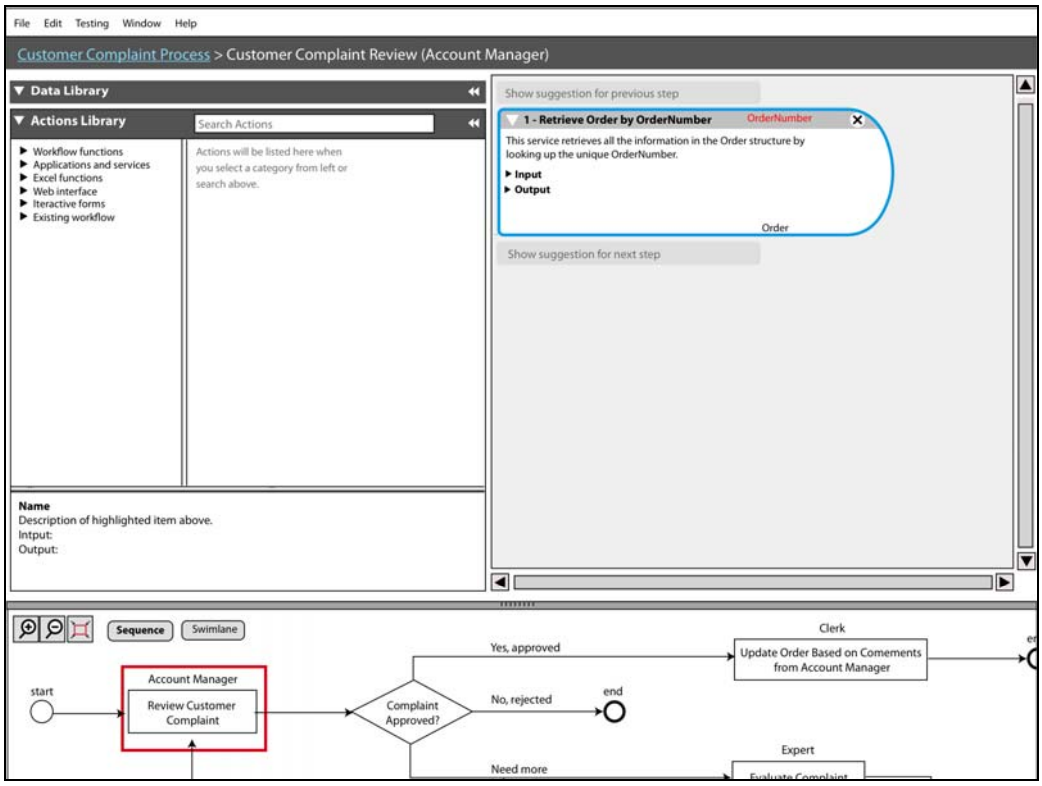

Figure 4-13. The Action Library (middle left) contains all the possible elements users would ever want to add to their workflows.

The Action Library integrated better with the workspace environment and provided better scalability. In iteration 1, the components were forced to categorize into either screens or services; anything that did not fit would require adding another button to the toolbar. We also observed from iteration one user testing that the pop-out template chooser caused minor disruption to user's task flow. In iteration 2, 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.

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 Action Library:

- The description box presenting details and functionalities about the highlighted action would assist users in selecting the most suitable action.
- The search box would release the users from the burden of browsing through the immense tree structure to find the desired components.
- The label of the taxonomy and actions should be in natural language and written in task-oriented fashion.
- The users could get suggestion on which action is suitable to be inserted. We imagined the suggestion could be based on matching data I/O type, common combination, and individual usage pattern. 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 4-14). The Data Library acted as a central registry of all the data output of all actions in the workflow. When a user wanted 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-n-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, 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.

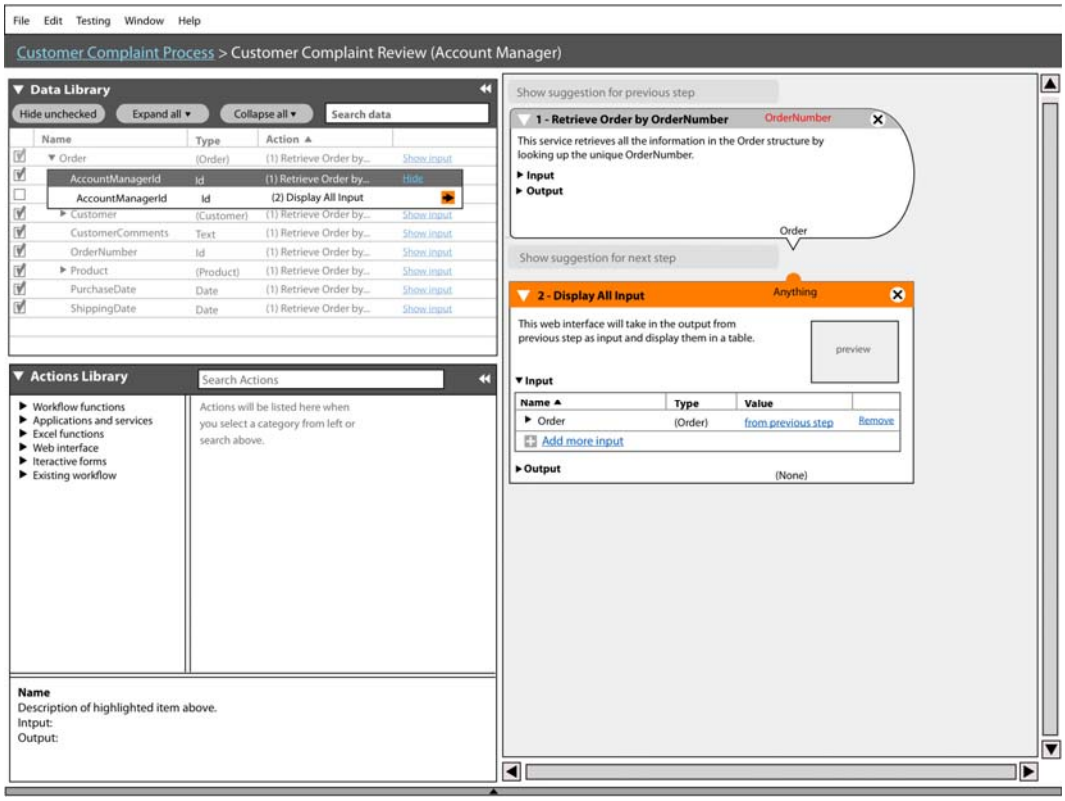

Figure 4-14. The Data Library (upper left) serves as a central data registry and the pivot point of defining data mapping through out a workflow model.

Finally, we replaced the zoom controls that were controversial in previous iterations with the Google Map navigation/zoom controls to experiment 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.

4.7.1.2. Testing Environment

Based on the iteration 1 testing results, our simulation concept blocked the user from building the mental model of how the system would function (see Appendix E). Because of this, we had rated changing this concept as top priority. According to user

feedback, we completely removed the prototyping concept in simulation, and changed the title simulation to testing. Once the user is in the testing view, a start-testing button triggers the interaction between users and screens or services.

Derived from classroom feedback, it was important to give users enough space to view the whole workflow to make selections before starting to test. From testing iteration 1, one user started to make random selections in the workflow, and the interface automatically divided into two split panels, with the workflow only displaying in the navigation thumbnail view (see Appendix E for detailed wireframes). From user interface heuristics, this clearly was inefficient use of screen space. However, in iteration 2, the workflow will be displayed full screen at all time for users to make selections before start testing is triggered. This improvement gives user more space to work with. Figure 4-15 shows the wireframe of the new testing environment.

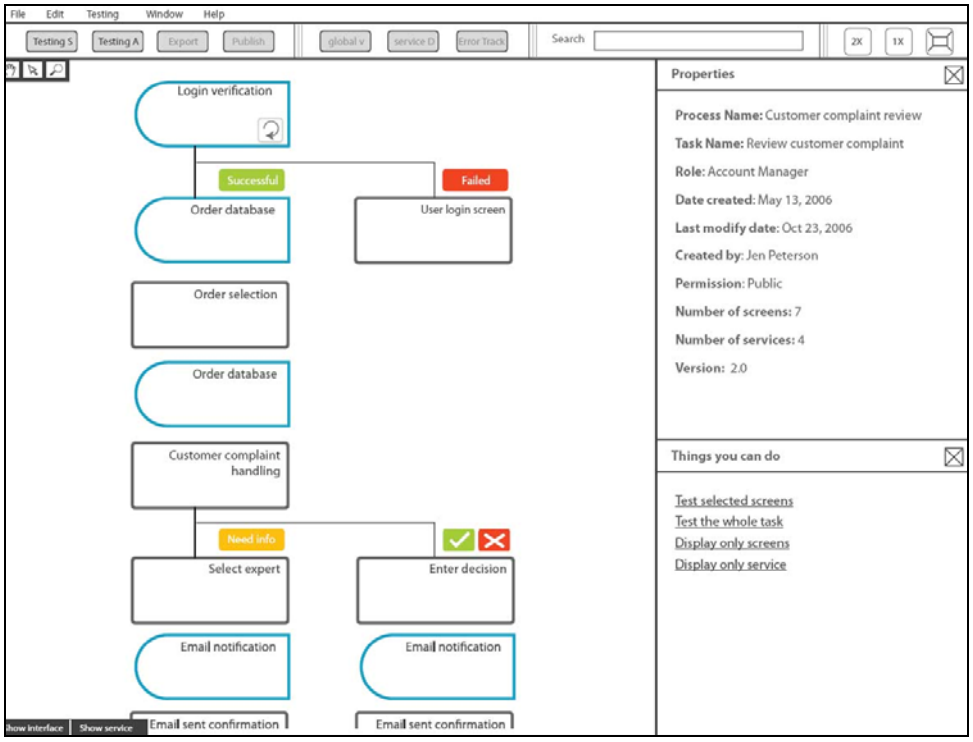

Figure 4-15 Testing environment

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). The selection methods we used in the previous iteration did not support path selection. However, in this iteration, we allowed users to select different path in the workflow, where a path is composed of many sub-paths. The system still provides auto-selection, such that once the user selects any component in the sub-path between two decision points; the whole sub-path will be automatically selected. Figure 4-16 show a screenshot of the new workflow selection method. We also decided to remove the numbering system, as this lead to confusion during testing. On the other hand, the removal of these numbers also prevented human errors such as mis-typing, etc.

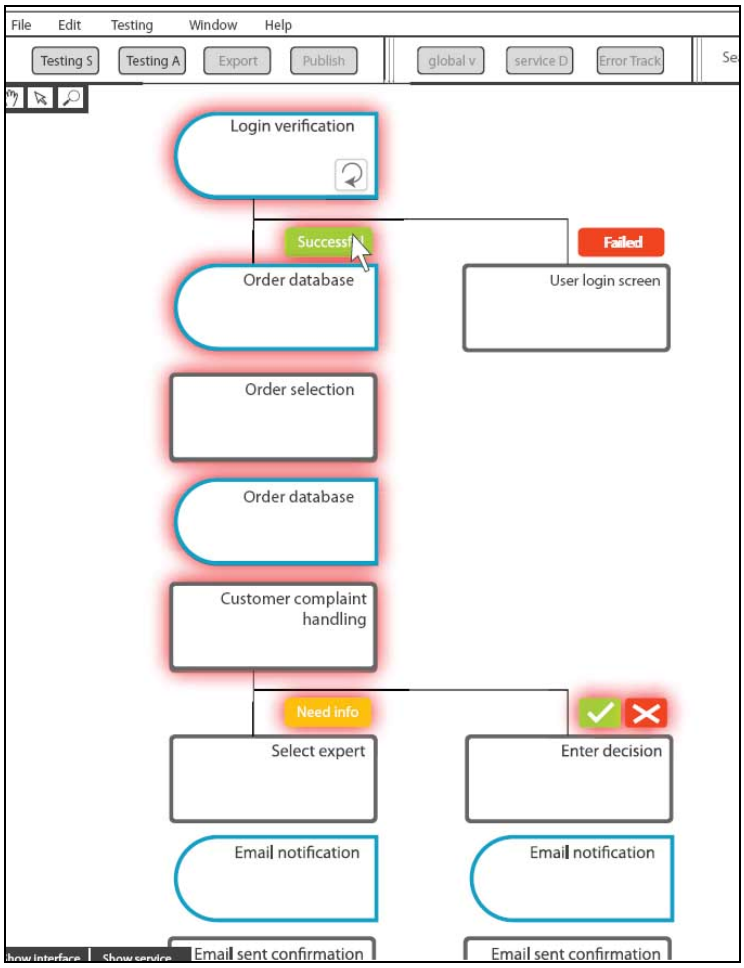

Figure 4-16. Testing environment - workflow selection

Given the nature of how business processes are designed, cross-functional calls are very commonly used (see Appendix E for details). As we proposed in iteration 1, some service calls cannot be complete because of broken or inactivated services. Multiple users wanted to be able to deactivate service calls if they know the call cannot be completed. As a result, the system does not need to waste time calling the service and return a failure error. In this iteration, we decided to add a dialog box popup; that displays all the required service calls in this workflow, where each is presented next to a checkbox. User can select which services they want to call before start testing. We suggest that the dialog box should automatically popup right after user presses the start-testing button. Figure 4-17 shows a screenshot of the service activation screen.

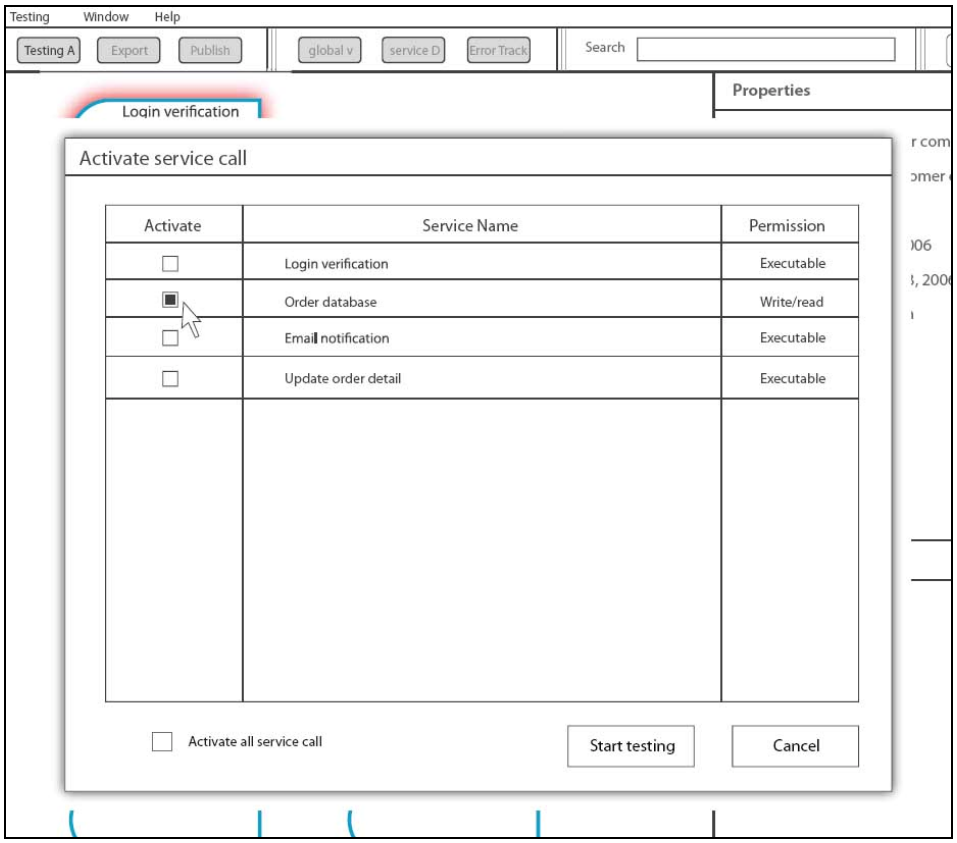

Figure 4-17. Testing environment - service activation

One new feature added in this iteration is letting users set environment variables. We learnt from our user research that is essential to run testing on data from different times (see Appendix E for detailed user feedback). Once the user has started testing, this option is displayed in the form of a popup box. Users can set the time; data and year the test should be run.

In iteration 1, we designed service dependency, which shows the data flow dependencies to help user gain a better understanding of the process. Figure 4-18 shows the screenshot of the designed service dependency feature. During testing, users showed excitement about this feature, but still felt the information we provide could be more helpful. They suggested it would be more valuable if it could show the actual data mapping, such as which field maps to which input data and vice-versa (see Appendix E for detailed user feedback). In this iteration, we improved the service dependency view to show data mapping, it displays exactly how the input and output data is mapped in the current service or screen. Figure 4-18 shows the iterated service dependency.

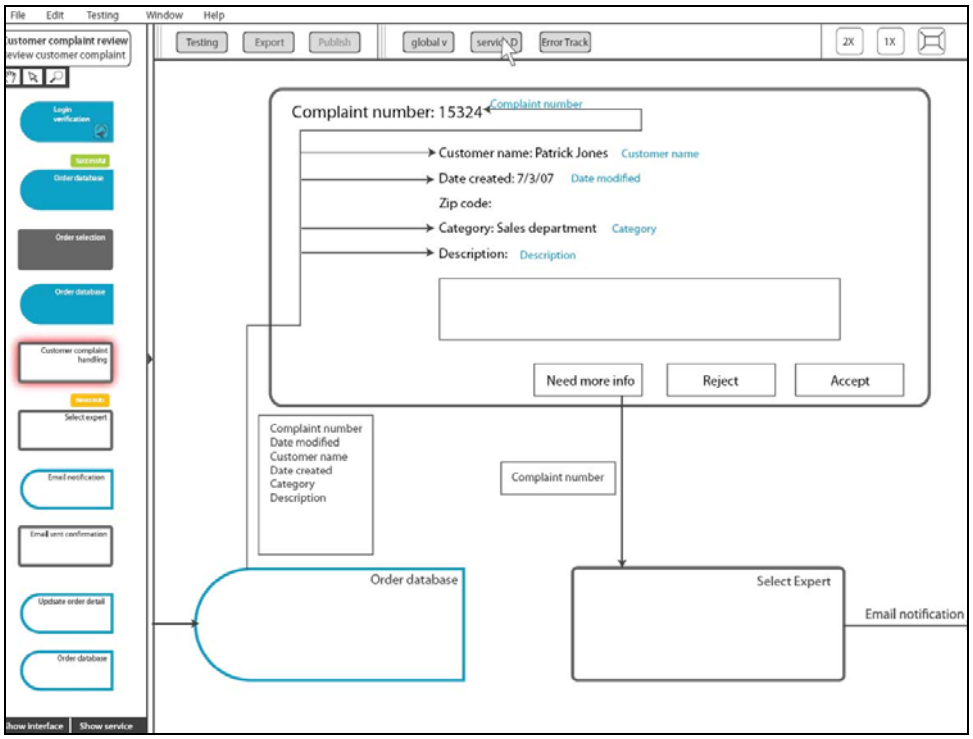

Figure 4-18. Testing environment - service dependency

Another major change was the decision to display service information as an individual screen during testing. In iteration 1, our system only illustrated user interface screens for interaction; service information was not available for users to view. Since the initial purpose of testing was to imitate real world web application interaction, we assumed that end users do not see the service information. During testing, multiple users conveyed the need of viewing service information; users in general like the interaction level, but for testing purpose, they still looking for more in-depth information (see Appendix E for detailed user feedback). We altered our design by adding the service information screen, using which users can insert required input data and view necessary details for each service. Figure 4-19 shows the screenshots of the service information screen.

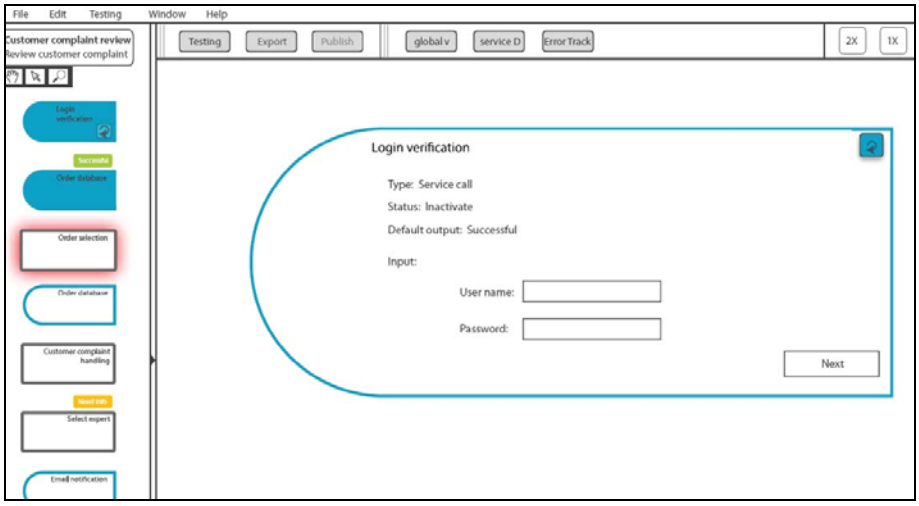

Figure 4-19. Testing environment - service information screen

As many users were not so fond of demo video studio, we decided to drop this part (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 is 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 created the new function in testing view, which allows users to export all the screens as screen shots into PDF. We strongly believe that users can use these images to generate tutorials and other requires documents.

Moreover, based on user request, we enabled free examination of any screen and service at anytime as users preferred. Since the option of entering customized service data was already there, this idea was an extension of that.

4.7.2 User Testing on Paper Prototypes

Similar to the user testing we have done for iteration 1, we performed Think Aloud Studies with four (4) business process experts to evaluate this iteration. 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 visualization. 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 revolves around learning how users react 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 users find iterated service dependencies provide enough information?
- Does workflow selection make sense to user?
- Do users find export screenshots function useful? What else they would like to see on the exported screenshots?
- What other service information is needed?
- Does service activation dialog box support user need?

The complete collection of direct verbal feedback from the user testing sessions of iteration 2 can be found in Appendix F.

4.7.3 Takeaways

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

4.7.3.1. 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 is a human step and which is a system step.

The users also expressed positively about the rigid workflow construction opposing 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 not 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."

4.7.3.2. 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.

4.7.3.3. 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.

4.7.3.4. 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.

4.7.3.5. Understand where you are in the process

Users often said that they got lost in the testing view, as they had no clue where they were in the process. In our prototype, we propose that user need to select a single task first and use testing environment to test the workflow inside that task. We provide navigation view for the workflow to help users understand which screen they are currently working on in relative to the workflow. Although we do have text display of which task you are currently testing, users did not perceive that

information, rather they prefer to view that information more visually. Users suggested that another small navigation of the process view might be helpful.

4.7.3.6. Information for exported screen shots

We received positive feedback on the exporting screen shots function. User suggests that to make this function more extensive, adding contextual information for each screen on the screen shots would be extra helpful. Such as input parameter, screen property, etc. This can act as a new add-ins function for export screen shots, which can be turn on or off according to user preference.

4.7.3.7. Test the whole process

Our system enables users to perform testing on each task workflow individually, but not multiple tasks at once. In other words, the system does not allow users to test an entire process. However, user 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. It serves as a good indication that we might need to extend the testing ability of the system.

4.8 Hi-Fidelity Prototype: Iteration 3 (P3)

After two rounds of paper prototyping, we gained an in-depth understanding of user needs. For this iteration, we created an interactive 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).

4.8.1 Prototype and Design Rationale

Based on classroom feedback, our system lacked consistency in display style, which included the tool bar and its location, navigation view, color scheme and layout style. In this iteration, as we had most of our functionalities in place, we decided to concentrate on the aesthetics and usability of our prototype.

Our prototype includes two different toolbars; one for navigation manipulation and the other is a system toolbar. The system toolbar contains tools that help users to accomplish a task, while a navigation toolbar helps user to navigate in the canvas. Following the traditional Windows style, we decided to put the system tool bar on the top. The navigation manipulation toolbar was put vertically on the left hand side of the screen, so that there is a clear distinction between the two toolbars.

We also spent time on picking out appropriate color schemes for our system. Based on human perception theory, we learnt that dark backgrounds do not disturb the user from focusing on the main task. Thus, we decided to use a dark color scheme.

One of our biggest design challenges was switching between different views in our system. Our current proposal was to have the contextual button in the menu bar. When the user is in Composer View, it would display testing. If it is in testing view, it will display composer. Users did not think such a button had the affordance to allow switching between views, because usually menu bars support sub functions. Thus, we decided to look into creating an intuitive mechanism of switching between different views. We decide to adapt the Microsoft Office 2007 round button style; three buttons which represented Process View, Workflow View and Testing View. These three button are been place on the top right corner which helps the user to differentiate them from other tools.

4.8.2 User Testing on Interactive Prototypes

We performed Think Aloud studies with three business process experts to evaluate the interactive prototype. 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 mental model, which we assumed in our hi-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 respond positively when encountering animation effects such as button highlighting on mouse over gestures or the animation of information panels.

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 has 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 including some members who had experience using JavaScript as well. A majority had also expertise using Flash, but had only novice knowledge in ActionScript programming. Most team members also turned out to have 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 be 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 in chapter 5.1.1. This development environment provides solid design tools [7] for the implementation of highly interactive software systems and gives almost unlimited flexibility when designing a custom look & 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 tools and technology.

5.1.3 Experience Report

The tools described in the chapter above enabled the team to let everybody 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 in form of XAML document. XAML, short for eXtensible Application Markup Language, allows the definition of executable user interface components without writing one line of traditional source code. How Designers and Developers collaborated to create an interactive prototype using the outlined development environment.

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 integrate well to be used in parallel to let both designers and programmers sit together and develop software artifacts in pairs. Team members found this type of collaboration a very encouraging experience.

5.1.4 Limitations

5.1.4.1. Limited ILOG License

The flow diagram design capability was not implemented from scratch for the JIGSAW prototype. A commercial program library from ILOG [9] was used to provide the diagram design functionality. 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.

5.1.4.2. 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 date rather limited. The Microsoft Expression Blend tools do provide timeline views of animations similar to Flash, but do not visualize the graphical elements which become animated. Instead, single properties of user interface components are animated, which reduces the animation to a bunch of changing variables.

5.1.4.3. Designing custom button styles in Blend

Buttons with a pleasing look and feel are easy to create on a first glance, but turn out to be complicated controls when having a closer look. 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 loose their focus while being pressed. Designing a custom button style in Blend is complex and therefore requires attention to detail and should explicitly be tested with users.

5.2 Prototyping Technology

5.2.1 Installing the prototype

5.2.1.1. Overview

This chapter explains in detail how to run the JIGSAW prototype. Both source code and binaries of the prototype are contained on the project CD-ROM which is handed-out along side this report. This CD-ROM also contains all the pre-requisites needed to run the prototype.

The prototype software artifacts won’t be available through the 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 don't 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.

To install ILOG Diagrammer, run setup.exe installer from the ILOG directory on the project CD-ROM. 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 ILOG directory of the project CD-ROM.

The development SITE license provided on the project CD-ROM is only intended to be used by Jonathan Gordon 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

To install the JIGSAW prototype, copy the program assemblies located in the "Prototype" directory on the CD-ROM to any folder on the local machine. The prototype can be executed by double-clicking the file "JIGSAWApplication.exe".

5.3 Final Design Solution

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 is consistently structured to provide a unified user experience in the different application modes. Figure 5-1 shows how the JIGSAW application window is structured into four major sections.

5.3.1.1. Menu Bar

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

5.3.1.2. Navigation Panel

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 control to switch between application modes to the right. The toolbar provides always functionality relevant to the work area whereas the control to switch between application modes remains the same at any time.

5.3.1.3. Work Area

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 5-2):

- **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 attached to the left side of the work area on demand.

- **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

The status bar indicates which process is currently being loaded in the JIGSAW application. The Sidebar does not yet provide additional information or contextual help (see chapter 5.4).

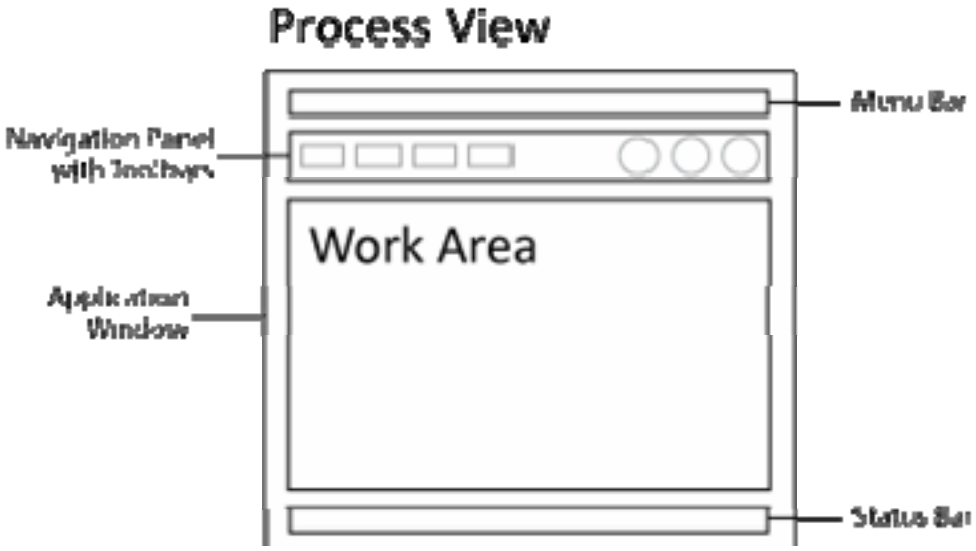

Figure 5-1: Static User Interface Structure

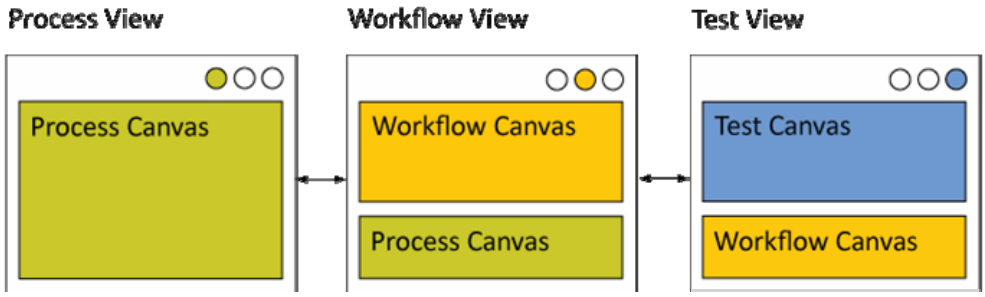

Figure 5-2: The three application modes of the JIGSAW prototype

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 5-3) in the upper left corner used to navigate in the displayed flow diagrams. It's functions are described in Figure 5-4.

Figure 5-3: Canvas Navigation Toolbar (Notice that this toolbar is vertically aligned in the prototype.)

Icon	Function Name	Interaction
Arrow	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.
Magnifying Glass (+)	Zoom In	Zooms into the currently displayed diagram.
Magnifying Glass (-)	Zoom Out	Zooms out of the currently displayed diagram.
Magnifying Glass Rectangle	Zoom into Area	Zoom into specified rectangular area. Requires mouse drag gesture while holding down the left mouse button.
Magnifying Glass 1:1	Original Zoom	Restore original zoom.
Magnifying Glass Cross	Zoom to Fit	Zoom to fit the diagram into the view port.
Hand	Pan	Drag the diagram while holding down the left mouse button.

Figure 5-4: Navigation Toolbar Functions

Overview

The design canvas which is of primary focus contains a small flow diagram overview (see Figure 5-5) 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.

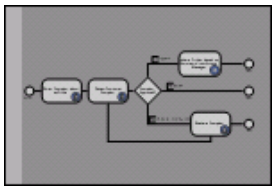

Figure 5-5: The diagram overview panel

Legend

The design canvas which is of primary focus contains a legend in 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 5-6 and Figure 5-7).

Figure 5-6: Legend for Process Canvas

Figure 5-7: Legend for Workflow Canvas

5.3.2 Process View

5.3.2.1. Overview

The Process View is the application mode initially shown when the JIGSAW prototype has started. It allows a user to design business process models based on basic elements defined in the standardized Business Process Modeling Notation (BPMN)

[6]. 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. When designing a business process model, the user creates manual and system tasks and connects them to form task sequences. Each process has at least one start point and end point. We distinguish manual tasks which represent offline work from system tasks which are supported by a composite application.

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 chapter 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 chapter 5.3.1.5). The horizontal toolbar contains functions to create elements of the process model. A legend (see chapter 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 (from left to right) manual tasks, system tasks, decisions, 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. Table 5-1 summarizes the functions available to manipulate a process model in the process canvas.

Icon	Function Name	Interaction
[Add Manual Task Icon]	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 Icon]	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 Icon]	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 Icon]	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 Icon]	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.

[Link Icon]	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.
[Arrow Icon]	Select single diagram element	Mouse Click on element on canvas
[Arrow Icon]	Multi-selection of diagram elements	Elements become selected which lie within or intersect with the rectangle defined through a mouse press and drag operation on the canvas.
[Arrow Icon]	Delete single or multiple elements	Select one or multiple elements and press Delete Key or choose Delete from the context menu.
[Arrow Icon]	Move single or multiple elements	Select one or multiple elements on the canvas, press the left mouse button and drag the mouse.

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

5.3.3 Workflow View

5.3.3.1. Overview

Workflow View 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 (when right-clicking a selected task). The work area becomes horizontally divided to show both the workflow canvas and the process canvas. The process canvas remains visible in order for the user to recognize that the currently displayed workflow is part of a particular system task of the business process.

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 chapter 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 no more be manipulated. The main toolbar (see Figure 5-8) primarily provides functions to open screen and service libraries, to create navigation and data flow connections and to delete selected elements from the workflow canvas (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!]. The blue headed screen shapes contain green and red input and output ports. The yellow headed

service shapes contain red input and output ports. The different colored ports are used to define two different types of links between UI screens and services. Green links are used to define navigational flow between screens while 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.

5.3.3.4. Main Toolbar

The first section in the top toolbar (see Figure 5-8) 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. More functions are provided in the second toolbar section. They are all highly experimental and not intended to be used in the JIGSAW prototype.

Figure 5-8: Workflow Design Toolbar

Icon	Function Name	Interaction
[Add Manual Task Icon]	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.
[Add System Task Icon]	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 Decision Icon]	Add Navigation Connection	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, a green connection is created in place. When dragging a connection end over a shape, connection points become highlighted.
[Add Start Point Icon]	Add Data Flow Connection	Mouse drag operation on canvas while holding down left mouse button. When releasing the left mouse button, a red connection is created in place. When dragging a connection end over a shape, connection points become highlighted.
[Arrow	Select single diagram element	Mouse Click on element on

Icon]		canvas
[Arrow Icon]	Multi-selection of diagram elements	Elements become selected which lie within or intersect with the rectangle defined through a mouse press and drag operation on the canvas.
[Red Cross]	Delete single or multiple elements	Click to delete the current selection.

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 5-9) and a user interface screen library. Both libraries provide re-usable software components for workflow design. 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 lower right section of the library panel. The upper right section shows the details of the currently selected components. By pressing either the [Add service] or [Add UI screen] button, a corresponding symbol is added to the workflow canvas. A symbol can be renamed on double-click.

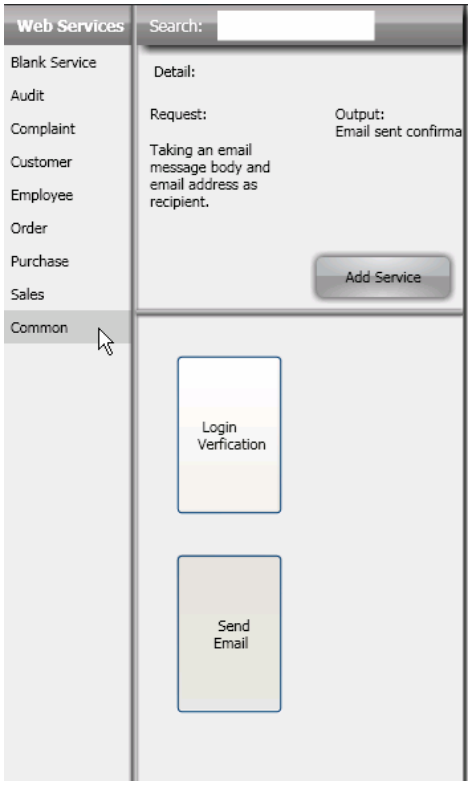

Figure 5-9: Library which offers re-usable components for workflow composition.

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.

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.

5.3.4.2. Main Toolbar

The first section in the main toolbar (see Figure 5-10) contains functions to start and stop a test, to publish a test case to stakeholders and to export screen and service details to a PDF document. The second section in the toolbar provides tools to analyze the run-time behavior of the composite application and the test results. Refer to Table 5-3 for a detailed description of these functions.

Figure 5-10: Workflow Test Toolbar

Function Name	Interaction
Start and Stop a test	Click to start or stop the test.
Publish a test	Not yet implemented
Export screens and service details to PDF	Not yet implemented
Change environment variables	Not yet implemented
Show data dependencies	Click to display data dependencies between workflow entities while testing.
Activate and Deactivate service calls	Once clicked, shows a dialog to select services.

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

5.4 Future Work

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 to 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.

5.4.1.1. 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. Deletion of selected elements has been implemented for the Process Canvas. More keyboard control should be implemented for all design canvas instances.

5.4.1.2. Testing choices

Before a test starts, let the user choose whether to test along the navigational flow or the data flow. As of now, the prototype only supports testing of the dataflow connections along the workflow.

5.5 Prototype Walkthrough

5.5.1 A usage scenario

In this prototype walkthrough, it is shown how business users can be empowered to design and implement composite applications without the need of extensive programming expertise using the JIGSAW application. The capabilities of this prototype are demonstrated by means of a real-world business process which defines how customer complaints are reviewed.

The following chapters walk the reader through use cases of the JIGSAW interactive prototype. The actor in these use cases is a business process expert who aims to improve the customer complaint by adding to the review process and changing one of its related workflows.

5.5.2 Running the prototype

Once successfully installed [reference to install description], the prototype readily contains this process and related test data. The reader is encouraged to run and experience the prototype while reading this chapter.

To run the prototype, go the folder containing the prototype assemblies [reference to install description] and execute JIGSAWApplication.exe. A splash screen (**Error! Reference source not found.**) indicates that the application is loading.

Figure 5-11: Splash Screen of JIGSAW prototype

5.5.3 Modeling a Business Process using BPMN

Once the prototype has finished loading, a process canvas is shown in the maximized application window. In general, the process canvas is used to display, manipulate and create business process models on a high-level of abstraction.

The process canvas already contains a model of the customer complaint review process for the purpose of this demonstration. In order to improve the process, the user wants to add another system task to the process which allows the review of the customer's complaint history prior to reviewing the new customer complaint.

The user first selects all process elements except for the “Enter customer complaint” 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 (see Figure 5-12). 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 task “Enter customer complaint” and task “Review customer complaint”. 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 5-13).

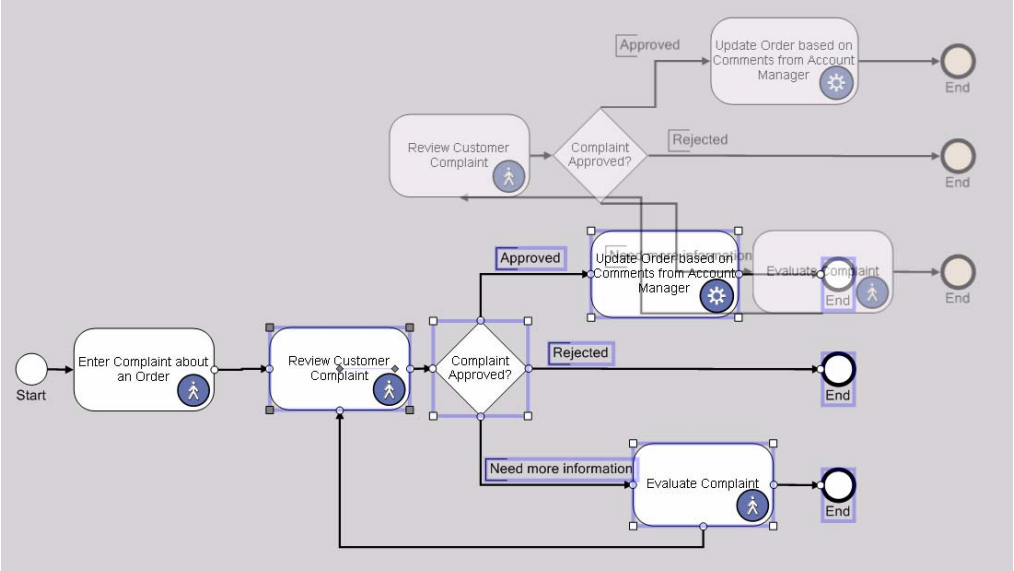

Figure 5-12: Moving multiple selected diagram elements on the process canvas by mouse drag gesture.

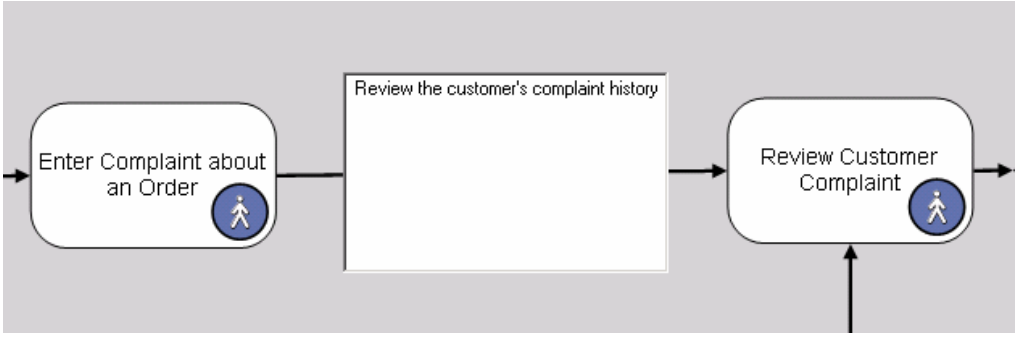

Figure 5-13: Enter a label for the new system task.

Thereafter, by clicking [Add Connection] button in the toolbar, the user can connect the new task with an existing task. The user again performs drag gestures on the canvas to create links and connect tasks.

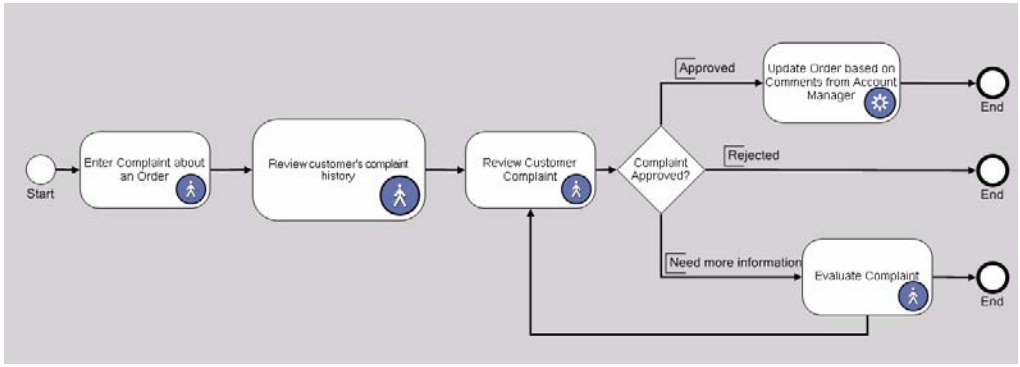

Figure 5-14: The finalized process diagram.

5.5.4 Creating a composite application workflow

The business process expert wants to extend the task “Review Customer Complaint” by adding automatic email notification to review experts. The user drills down into this task by right-clicking it (see Figure 5-15) and selecting [Edit this task].

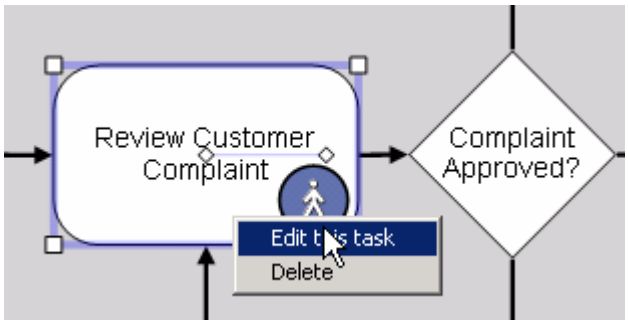

Figure 5-15: Right-clicking a task shows a context menu with functions to edit or delete this task.

The window now contains a second diagram canvas above the process canvas (Figure 5-16). This new work area, further referred to as “Workflow Canvas”, is used to edit the workflow behind a process task. The lower process canvas which previously allowed the user to edit the process flow has now changed to be read-only, even though the user can still navigate in the process model.

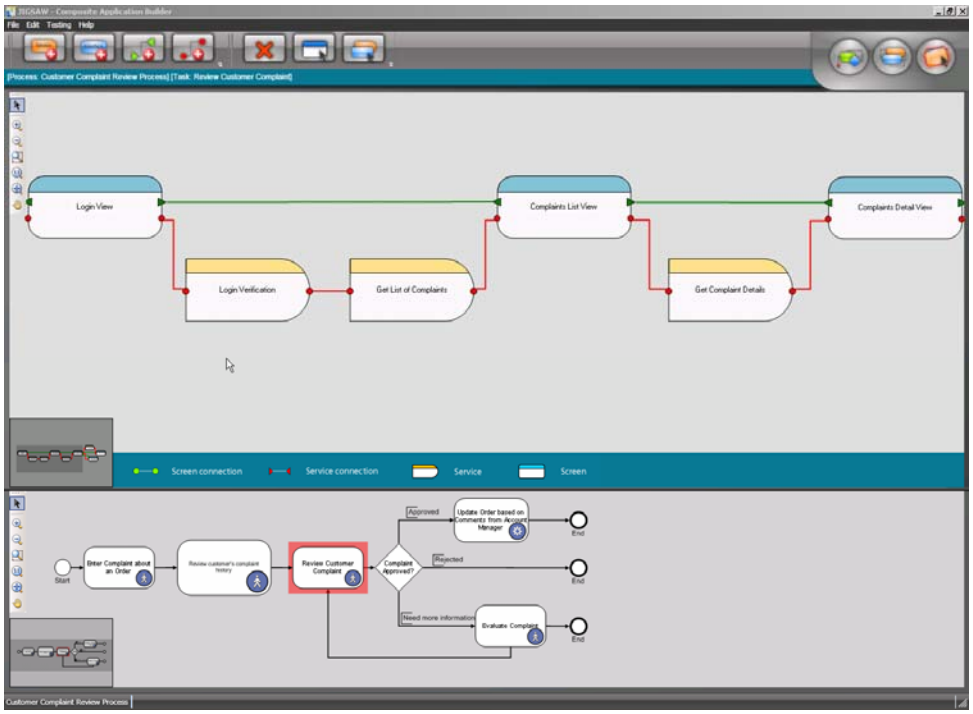

Figure 5-16: The workflow canvas with the read-only process canvas in the bottom of the window.

The Workflow Canvas shows part of the “Review Customer Complaint” workflow. It consists of two different types of symbols: UI 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 to the end of the workflow. The workflow and can be brought into view by panning the canvas to the left. The user therefore selects the pan tool from the vertical toolbar on the left side. The cursor changes to a hand symbol to indicate that the user can perform mouse drag gestures on the canvas to shift the current view port. The user shifts the workflow model to the left side of the screen 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 attached to the left side of the workflow canvas (see Figure 5-17).

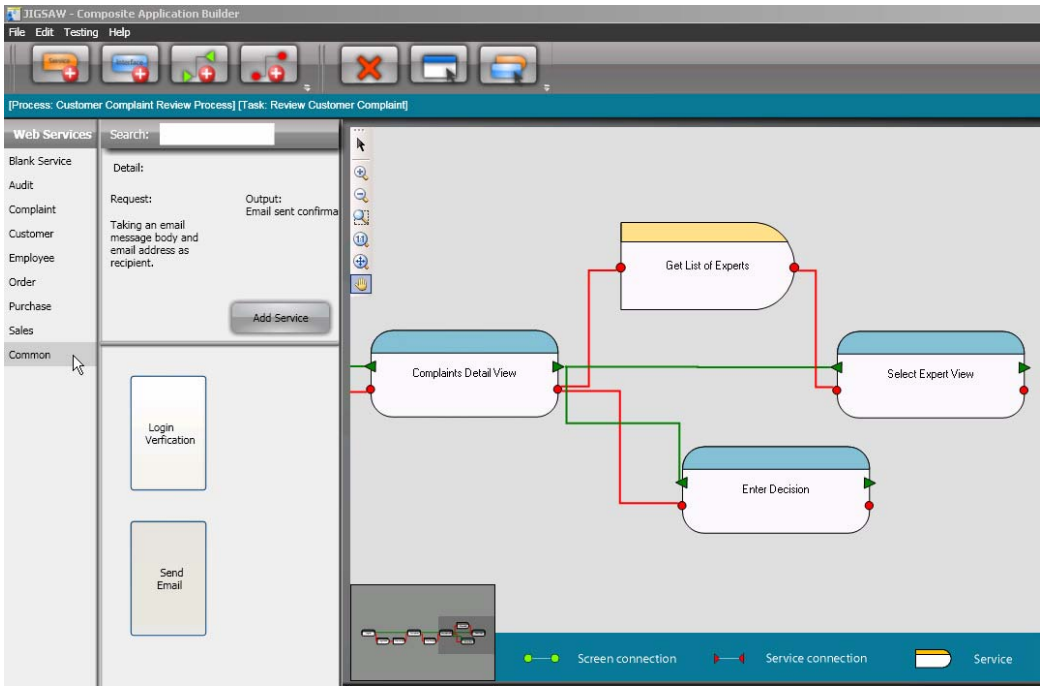

Figure 5-17: The Service Library showing common services

The “Send Email” service can be found under the Common service category in the first column in the Service Libray side-bar. The user clicks on [Send Email] to see the details of this service. The service can be added to the Workflow Canvas by clicking [Add Service] next to the detailed description.

The user also wants to add a confirmation to the workflow to indicate that an email has been sent. A confirmation screen can be selected from the UI screen library which is shown when clicking [Add UI Screen] button in the top toolbar of the window. An email sent confirmation screen can be found in the “Review” category of the UI screen library. The UI screen is added to the workflow canvas by clicking the [Add UI screen] button.

Both the new service and the new screen are not yet connected with the existing workflow. Use both the navigation flow and data flow connection tools in the top toolbar of the window to incorporate the new workflow elements into the workflow (see Figure 5-18).

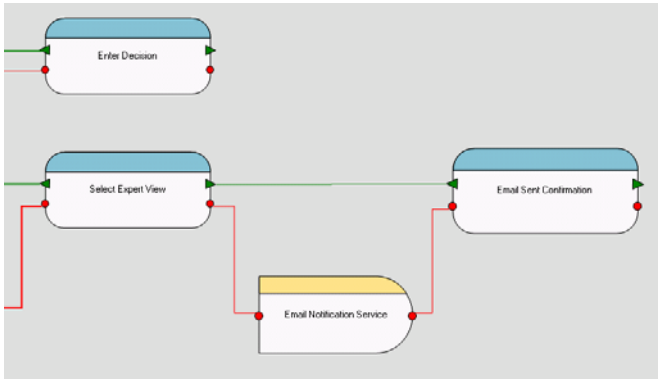

Figure 5-18: The new workflow elements integrated into the existing workflow.

5.5.5 Testing the workflow

Modeling a workflow from UI 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 test environment by clicking the right most navigation button in the upper right corner of the window. The test environment shifts the workflow canvas to the bottom of the screen to make space for detailed views on screens and service instances (see Figure 5-19). For each detailed view there are also properties (see Figure 5-20) to be inspected by the user. They can be accessed by clicking the [Properties] button attached to the upper right of the window.

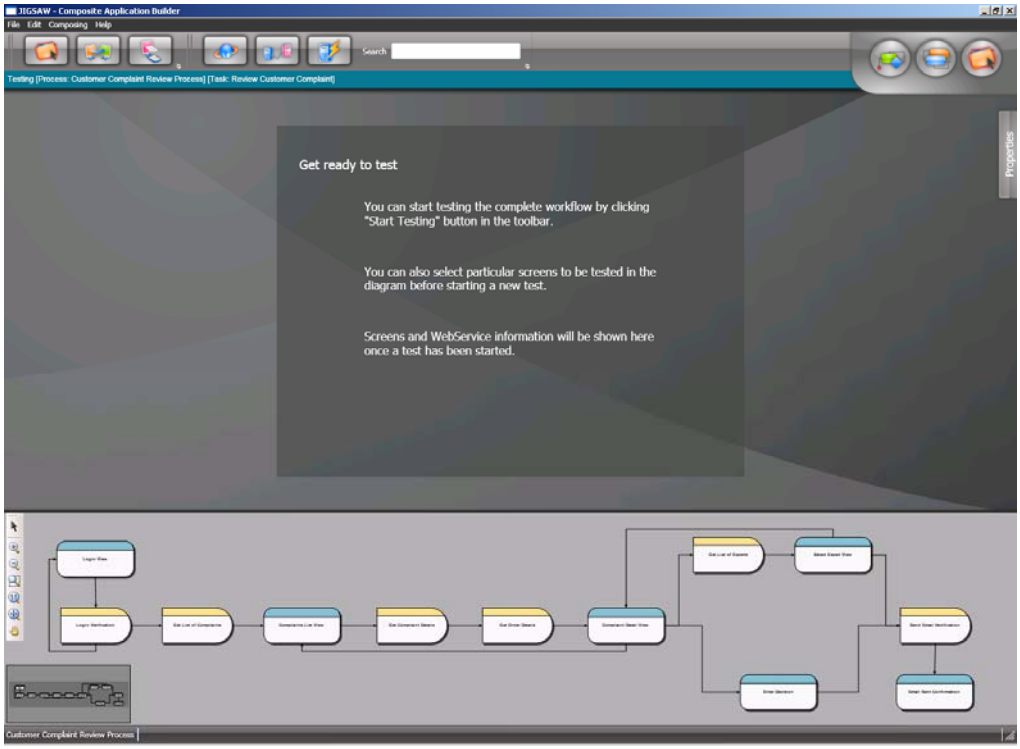

Figure 5-19: The workflow test environment before starting a test.

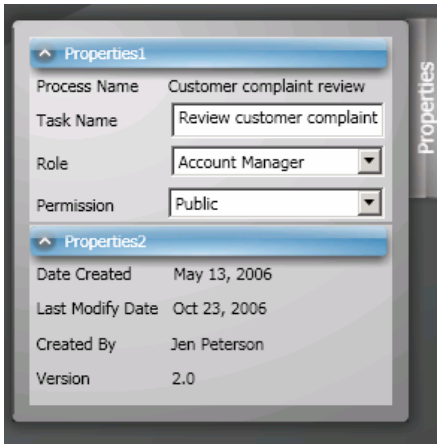

Figure 5-20: The property sheet provides detailed information about screens and services currently displayed in the work area.

The user can select entities in the workflow model to see details of both user interfaces and services. Screens are represented by showing the detailed user

interface elements (see Figure 5-21). Services are represented by showing input and output parameters plus details about the service implementation (see Figure 5-22). The workflow canvas shows the data flow among workflow entities by default. The user can switch between dataflow and navigation flow on-demand.

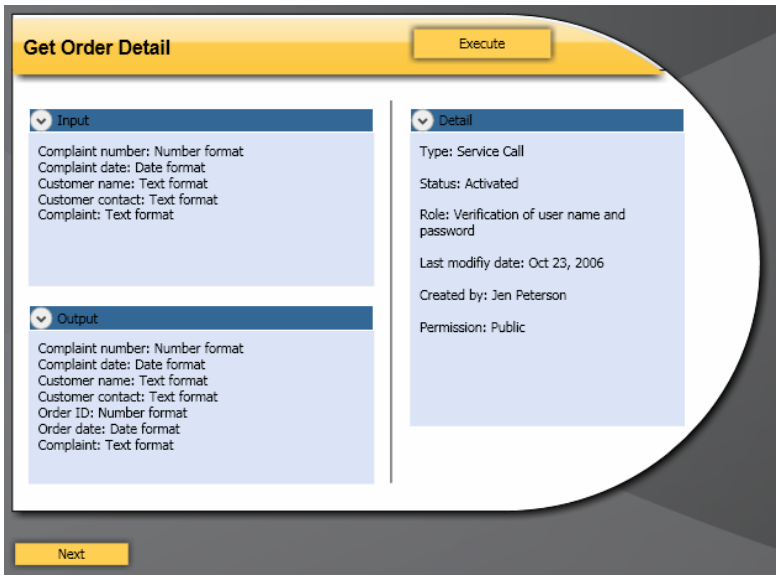

Figure 5-21: Service detail view in the workflow test environment.

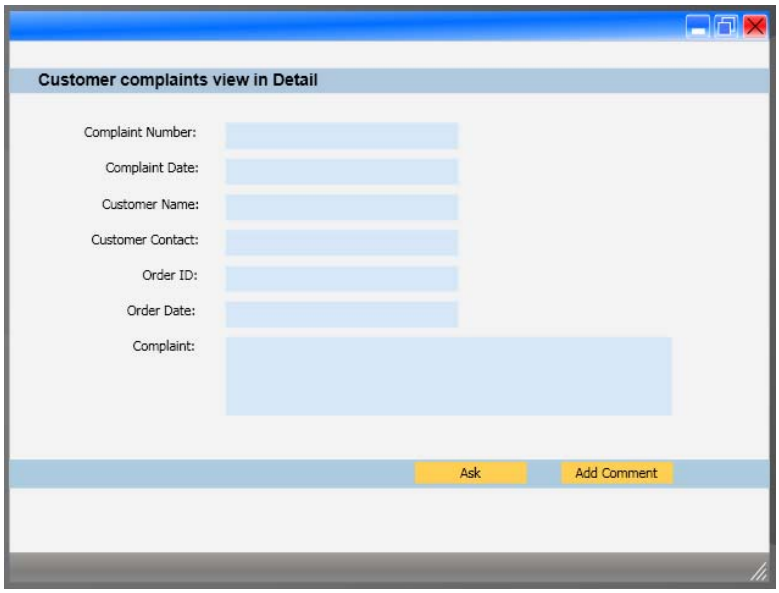

Figure 5-22: UI Screen detail view in the workflow test environment.

The user can start a test by clicking the [Start Test] button in the top toolbar of the window. A screen The first screen of the workflow is highlighted in the Workflow Canvas and the corresponding screen is displayed. The user interacts with the displayed screens to drive the test along the links shown in the workflow model.

Once a path through the workflow has been tested to the final screen or service, a test report is shown to the user (see Figure 5-23). 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.

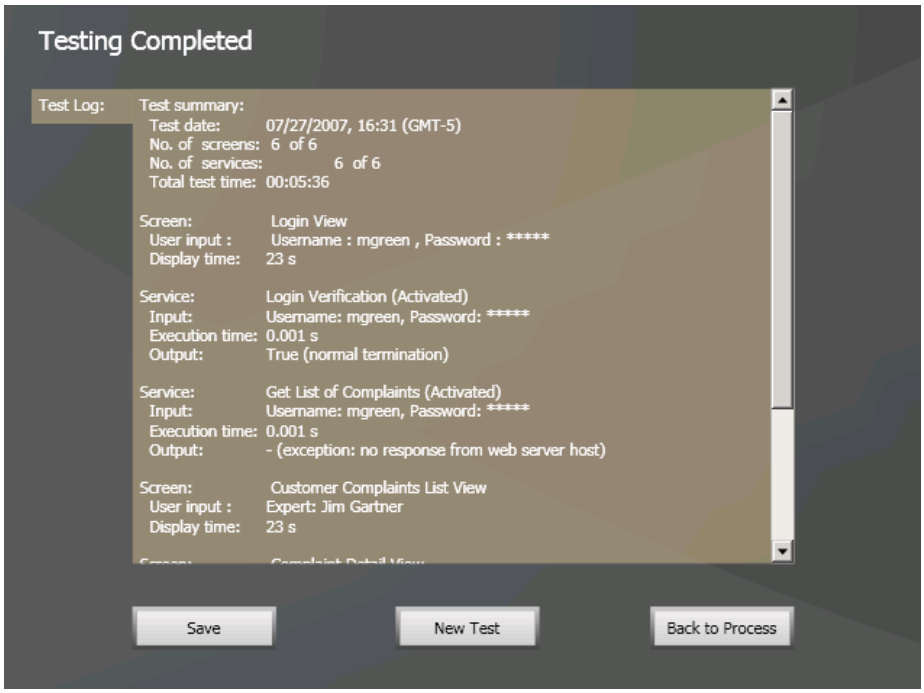

Figure 5-23: Summary report of a test case.

The user can either save the report to PDF, finish the test or go back to the process view. When the user finishes the test, the initial screen with instructions is shown again, but the test environment does still hold the test parameters previously entered. The user can like this explore the details of the test even after the test has been finished.

6 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. 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 project initiatives 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.

7 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.

8 Indexes

8.1 Index of Tables

8.2 Index of Images