

# Demonstration Description of the *s*View System

**Markus Bylund**

Swedish Institute of Computer Science

Lägerhyddsvägen 18

SE-752 37 Uppsala, SWEDEN

+46 18 471 70 53

bylund@sics.se

**Fredrik Espinoza**

Swedish Institute of Computer Science

Box 1263

SE-164 29 Kista, SWEDEN

+46 8 633 15 15

espinoza@sics.se

## Abstract

The role of the Internet is about to change again. After the hype of web portals we look forward to an Internet in which networked services dominate. Already today we can find services on the Internet in a large number of different categories: communication, on-line shopping, entertainment, enterprise services to mention only a few.

However, problems with the currently prevailing infrastructure for networked services, the World Wide Web (WWW), slow down this process. We argue that there are three main problem areas: limited support for service collaboration, producer dominance, and the poor support for service interaction in today's web browsers.

A Personal Service Environment (PSE) enables a rich interaction between networked services and their users. In this demonstration, we present the concept of a PSE in general and the *s*View system (which among other things implements PSEs) in particular. A PSE is a general interaction environment for improved interaction between networked services and their users. A PSE can store information about its user, but also execute and store service specific logic and data.

## INTRODUCTION

A PSE, which is private to an individual user, enables user interaction with networked services.

The PSE can store and execute service logic and data, and it can move from computer to computer as the user moves between access nodes within the network.

Preferably, the PSE is run on a computer under the user's immediate control (such as his/her workstation at work or personal computer at home), but if the user is not directly represented on the network, the PSE can run on a server that is dedicated to running PSEs.

The choice of protocol for interaction between service logic within a PSE and its user is open. The PSE can provide a range of channels for services to interact with their users

(whichever channel is available for the moment), e.g. HTML over HTTP, WML over WAP, ASCII over SMS, GUIs via SWING, etc.

## IMPLEMENTATION

The *s*View system is a complete system for user-service interaction that implements PSEs. The system is authored exclusively in Java 1.2, which brings several advantages. The choice of implementation language makes the system platform independent. The popularity of Java makes it easy to reach a broad user group, and finally, Jini technology can be used for naming, searching and distribution of service components.

The *s*View system constitutes a very thin layer of infrastructure between the service providers and their users. Nothing is assumed about communication protocols between base services and service components within a PSE, or about interaction protocols between service components and the devices that render service presentations. Similarly, no languages, interpreters, or interaction styles are forced upon service components that wish to collaborate within the PSE.

The thin properties of the *s*View system are a deliberate design choice. We want the system to be specific in its overall goal (to enable user-service interaction), but open when it comes to methods of achieving the goal.

## SAMPLE SCENARIO

Below we describe a usage scenario that illustrates some features of the PSE.

*A man is about to make a business trip to a foreign city. Using his Personal Service Environment, he locates an electronically mediated travel agency and initiates a dialog with it.*

*The travel agency uploads a travel service component to the user's PSE.*

*Once in the PSE the travel service receives the man's instructions, via a standard graphical user interface (GUI), to make a flight and hotel reservation for his planned trip.*

*Then the man turns his attention to something else and leaves the office. But before doing so, he lets*

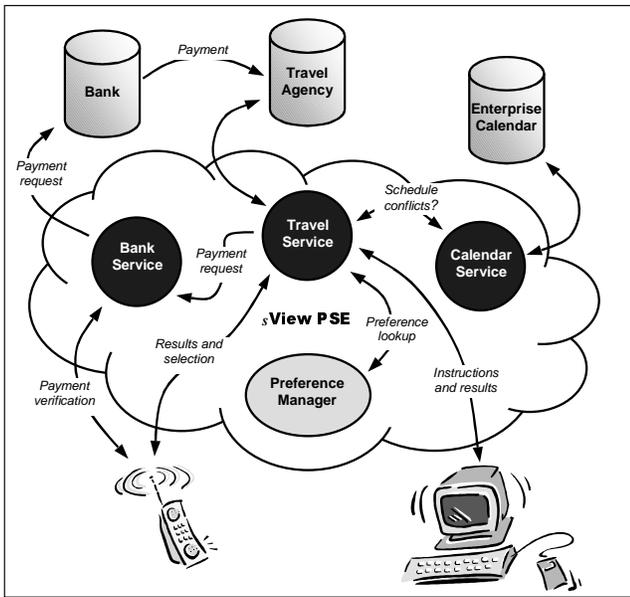


Figure 1. A schematic overview of the described scenario.

his PSE know that he is no longer available via his desktop computer but rather via his cellular phone.

The travel service now makes use of a number of information sources in order to accomplish its task. It searches the PSE for a preference manager and asks it about its client's complete name and address, as well as his seating and smoking preferences. It also locates a calendar within the PSE and checks when the man must be back and if the trip conflicts with any of his other appointments.

Having collected all background information, the travel service turns to its base service trying to find an appropriate flight and hotel. The service finds three alternatives that all match the man's request, preferences, and schedule.

The travel agency is now ready to get back to the client with the result of the search. However, since the man is no longer available via the desktop computer, the service contacts him via his cellular phone. The man, now on the train on his way home, selects one of the alternatives and instructs the travel agency to go ahead with the reservation.

The travel service accepts the request and starts searching the client's PSE again, this time for a service that provides payment. One of the man's services, a bank service, is willing to provide payment, but only after a confirmation by the client (this is also done through the interface of the cellular telephone).

Having everything that is needed, including payment, the travel service now executes the man's request by instructing its base service to buy the flight tickets and make the hotel reservation.

The above scenario illustrates three important aspects of the PSE:

- The PSE allows, and actively supports, service collaboration to take place within the environment.
- Since the infrastructure does not *require* HTML/HTTP, interaction through a number of different devices without loss of interaction state is possible.
- The user shares personal information (such as preferences) with a special purpose component within the PSE (the preference manager). This makes it easy for the user to add, change, inspect, and retract information without having to contact every service that is used. At the same time, services have a central source for such information for every user.

### DEMONSTRATION

The scenario described above constitutes the core of the demonstration; nearly the whole scenario will be presented<sup>1</sup>.

In addition to what the above scenario illustrates, the following features of the sView system will be demonstrated:

- Service components can be uploaded to the PSE for local execution.
- A PSE with all its service components can follow the user in the network by moving between network nodes.
- The use of three different interfaces for interaction with the same service (HTML/HTTP, WML/WAP, GUI/SWING).
- The interaction state can be preserved between usage sessions and throughout the transfer of interaction between devices.
- Users' control over personal information can be increased by centralizing management of the information to a component within the PSE.
- Users' control over service collaboration can be increased by allowing the collaboration to take place within the PSE.
- The network dependency can be reduced when interacting with networked services by allowing services to upload and execute service components in users' PSEs.

### CONCLUSIONS

A Personal Service Environment (PSE) enables a rich interaction between networked services and their users. We have introduced the concept of a Personal Service Environment as a solution to many of the problems with using the WWW for mediation of networked services.

The main contribution of the PSE is that it increases its user's control of networked services. Services are offered support for spontaneous collaboration between peers.

Future work includes using the sView system as a platform for further research in the fields of service-service collaboration and personalization of services.

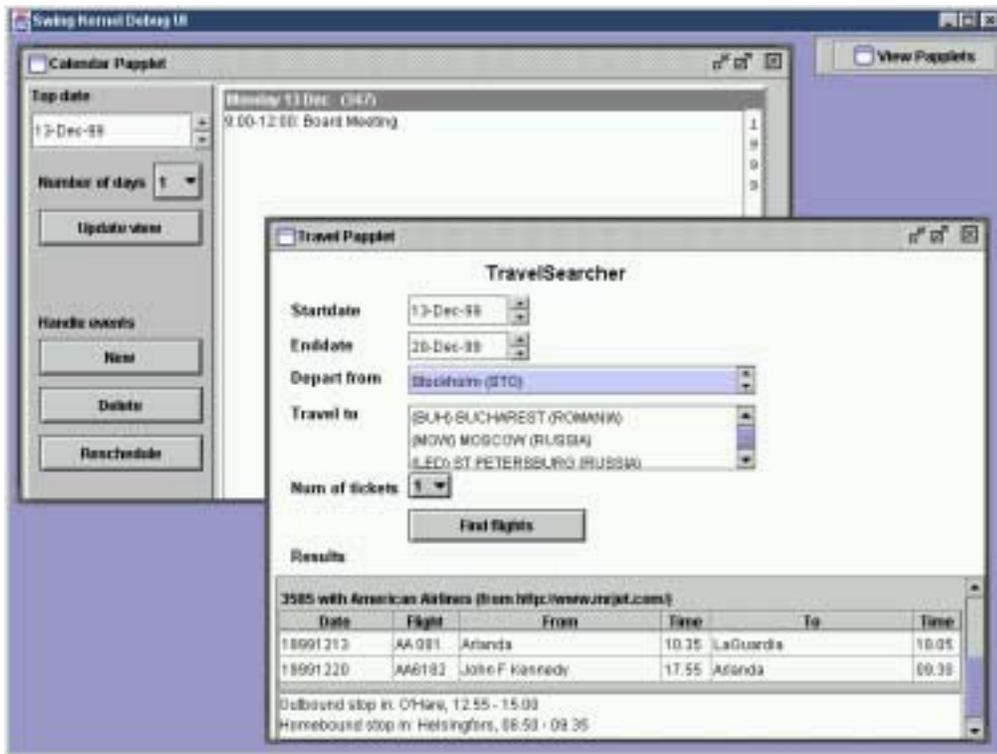


Figure 2. Screenshots from interactions with service components in the sView system. The calendar service and the travel service from the scenario above interact with the user through a SWING GUI. The payment service interacts via a cellular telephone.<sup>ii</sup>

<sup>i</sup> Two details of the scenario can not be demonstrated. Firstly, the travel service is not capable of making hotel reservations. Secondly, since neither of the travel agencies that we use behind the travel service accepts Jalda payments (See <http://www.jalda.com/>), the actual purchase must be simulated.

<sup>ii</sup> The snapshot of the cellular phone is taken from the Nokia WAP Toolkit 1.2 that is used to simulate WAP enabled cellular phones in absence of real ones ( See <http://www.forum.nokia.com/developers/wap/wap.html>). WAP phones are available for European 900MHz/1800MHz digital networks, but not yet for American 800MHz/1900MHz digital networks.