The XVC Framework for In-Vehicle User Interfaces

Jonathan Munson
IBM T.J. Watson Research Center
Young Ju Tak
IBM Ubiquitous Computing Laboratory, Seoul
Outline

1. **Motivation**
   - How to bring the qualities of the Internet-based marketplace to the telematics domain?
   - XVC is the UI framework part of our answer to that.

2. **The TOPAZ Vision**
   - Marketplace vision and high-level application model

3. **The XVC Model**
   - Application Composition
   - Viewer Composition

4. **XVC Demo**
Many Valuable Telematics Applications

- Traffic information
- Dynamic navigation
- Road warnings
- Driver Status
- Traffic & navigation
- Mobile office
- e-Intelligent driving
- e-Intelligent driving
- m-Commerce
- Information services
- Multimedia entertainment
- Emergency & safety
- Emergency assistance
- Door locking / unlocking
- Remote diagnostics
- Messaging
- Collaboration
- Colleague tracking
- PAYD Insurance
- Road Metering
- Points of interest
- News
- Friends & family
- Infotainment
- Pervasive gaming
- Computer, games
...But For Application Providers, a High Cost

- In current marketplace, cost of entry is high
  - End-to-end systems are expensive to purchase and operate
  - Difficult to justify for single applications
- In a more ideal marketplace:
  - Cost of entry for application providers is low
    - Applications are easy to develop and deploy
    - Encourages innovation and experimentation
  - Attracts many application providers, large and small
    - For both popular applications and “niche” ones
  - Application providers compete for customers on features, quality, price
    - Even basic applications become interesting…
    - Consumers have choice
The TOPAZ Marketplace Vision

- Using TOPAZ, Telematics Infrastructure Providers factor out the telematics-intensive parts of telematics and ubiquitous applications, and offer them as services to any application provider.
- Infrastructure providers and application providers are independent business entities.
  - Applications are separated from the platform operated by the infrastructure provider, and operate independently from the platform and from each other.
  - TOPAZ platform services are accessed via Web-service interfaces. One platform operator serves many application providers.
  - Cost of infrastructure is shared by many apps; app providers can enter the marketplace at low cost.
- A range of specific business models are possible within this basic model.
TOPAZ Application/Platform/Client Architecture

- TOPAZ basic application: applications run on servers and present user interfaces using viewers
- TOPAZ defines a set of core telematics services
  - On-demand data acquisition from client devices; content push to client, managed for access by multiple application providers; rule-based spatiotemporal event detection
  - The XVC framework enables users to receive multiple push-based services simultaneously
- TOPAZ utility-computing functions
  - Adding, removing, updating applications; application providers; subscribers & devices
  - Managing the requirements of diverse applications; Managing system load due to large numbers of subscribers; managing load on individual clients; managing load on individual applications
  - Service metering, monitoring, and diagnostics
XVC Motivations & Requirements

- Enable multiple, independent application providers to deliver user interfaces to a driver’s in-vehicle display, simultaneously.
- Take into account the limitation of user’s device like small screens and limited computing resources.
- Take into account the usage context:
  - Attention–limited users
  - Information–driven services rather than intensive information–processing services.
- Enable application providers to easily develop, deploy, and maintain user interfaces for Web–based applications.
XVC Overview

XVC is a framework enabling Web-based applications to present interactive user interfaces on in-vehicle clients.

- **Document-oriented model**
  - XVC offers a document-oriented user interface model, in which, like the Web model, applications present user interfaces through documents rendered by universal interactive viewers.
  - No need to develop custom clients.

- **Viewer composition—an extension of the Web UI model**
  - XVC introduces telematics-oriented viewers, each tailored to different content medium.
  - Its composite document model composes user interfaces by delivering documents to multiple viewers.
  - Inter-viewer interactions are enabled through event handlers.

- **Application composition**
  - XVC composes the user interfaces of multiple applications into a merged user interface
  - Purpose is to enable “glanceable” displays as much as possible.
Document-Oriented User Interface Model

*XVC complements the TOPAZ application model with a document-oriented user interface model, in which, like the Web model, applications present user interfaces through documents rendered by universal interactive viewers.*
Applications compose user-interfaces by sending to XVC compound documents that contain documents for one or more viewers. The viewers are arranged on the screen according to a layout specification received by the client when the application is subscribed to.
Application Composition

*XVC’s application composition model enables multiple applications to share the screen simultaneously.*
# TOPAZ Client Components (8) : Layout Examples

<table>
<thead>
<tr>
<th></th>
<th>Map Viewer</th>
<th>HTML Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>TOPAZ Client Components (8)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="utf-8"?>
<layout id="1">
  <row size="70%">
    <col size="60%" rowspan="2">map</col>
    <col>html</col>
  </row>
  <row>
    <col>buddytrack</col>
  </row>
</layout>
```

<table>
<thead>
<tr>
<th></th>
<th>Map Viewer</th>
<th>HTML Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom (BuddyTrack)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="utf-8"?>
<layout id="1">
  <row size="50%">
    <col size="50%">map</col>
    <col>html</col>
  </row>
  <row>
    <col>buddytrack</col>
  </row>
</layout>
```

<table>
<thead>
<tr>
<th></th>
<th>Map Viewer</th>
<th>HTML Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Custom (BuddyTrack)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```xml
<?xml version="1.0" encoding="utf-8"?>
<layout id="1">
  <row>
    <col size="50%">map</col>
    <col>html</col>
  </row>
  <row>
    <col>buddytrack</col>
    <col>chatting</col>
  </row>
</layout>
```
TOPAZ Client UI Design (1) : Overview

- Dock Viewer
- Main Screen
- Map Viewer
- Application Menu Viewer
- Ticker Viewer
- HTML Viewer
- Layout Manager
- Hard Button Simulator