FORUM NOKIA

Implementing Trucking Logistics with Nokia Handsets

Version 1.0; July 18, 2003

Java™/XML

NOKIA

Contents

Change History

18 July 2003	V1.0	Initial document release

Disclaimer

Figure 1:The information in this document is provided “as is,” with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

License

A license is hereby granted to download and print a copy of this specification for personal use only. No other license to any other intellectual property rights is granted herein.

Implementing Trucking Logistics with Nokia Handsets

Version 1.0; July 18, 2003

1 Introduction: On the Road Again

Traditionally, a trucker's job is a lonely one. For unending miles and hours, he's alone in his cab, traveling from place to place — out of touch with his family, his suppliers, and his customers. Thanks to one company, that situation is beginning to change.

Somewhere in mid-America, a truck driver pulls his rig into a roadside stop. While the truck is being fueled, he pulls out his Internet-equipped Nokia mobile phone and taps in a few numbers. While the truck driver sips his first cup of coffee, he checks his business accounts, informs the shipper of his present location and expected arrival time, and arranges to pick up a new load for his return trip.

[A] [B]

Figure 2: From anywhere that a mobile phone can reach, Landstar truck drivers (Business Capacity Operators) can log into the system [A], locate and select their next load, and view the job's relevant parameters [B].

This is the typical daily experience — today — for the owner-operators who contract to Landstar System, Inc. (http://www.landstar.com), a Fortune 1000 firm that provides contract logistics transportation services across the United States. Since fall 2001, Landstar's B2B customer relationship management (CRM) wireless application has delivered information faster to the 8,800 independent truck drivers who contract with the company. The better access to dynamic data gives the truck drivers, called Business Capacity Operators (BCOs), a competitive advantage and a measurable increase to their bottom line.

Landstar's development process illustrates several user interface lessons that can help wireless developers. This white paper looks at the company's design and development methodology, its technology choices, and how the company solved the problems it encountered.

1.1 The Business Process

Landstar had a Web-based CRM application in place before it began its wireless development project in January 2001. A BCO equipped with a laptop computer and modem could dial in to learn his account balance, find an available load for his next contract, and so on. But to get there, the driver had to set up a computer, find an ISP with local access, and dial in — often on a relatively slow connection. That meant waiting for a rest stop, if not an overnight hotel stay. The access inconvenience limited the site's usefulness.

Landstar's new wireless application lets a BCO use a menu-based system to log on and get the needed data without concern for dial-up lines or the distance to the next truck stop. With a microbrowser-based phone, pager, personal digital assistant, or other wireless device, the BCO connects to the company database to retrieve the needed information from anywhere there is mobile phone coverage. The mobile phone is, by far, the most commonly used device, because so many truck drivers have other reasons to carry one. Because the software works with any mobile phone with microbrowser capabilities, the BCOs can choose the model they prefer.

Currently, between 2,700 and 2,800 of the 8,800 BCOs use the wireless system. That's an impressive adoption rate, because only about 35 percent of North America has mobile phone coverage. Also, some owner-operators have predictable and regular routes, so not all of them need to provide or receive up-to-the minute updates.

Landstar's developers couldn't design a theoretical application based solely on what they thought would appropriate to include. Several usage criteria were built in. For instance, because the BCOs are contractors, not company employees, Landstar couldn't mandate which brand or model of phone the driver might choose. Even if the company did attempt to legislate a technology choice, limitations in wireless service providers and coverage across the country would make such a requirement untenable. Although the initial project plan included a Nokia WAP Gateway, it was dropped once the decision was made to keep the Landstar system device- agnostic.

As a result, Landstar adopted an open architecture platform. The company's code relies on a wireless application gateway interface called BlueMoon, acquired from Airtuit Inc., which was previously PhoneOnline.com.

The developers also created a list of critical system requirements, which emphasized ease of use. The application had to minimize text entry on the phone. Response time had to be optimized to a maximum of four or five seconds from submittal. The application needed a consistent UI, and navigation had to be intuitive. Those criteria drove the initial development. Two years later, the application's designers say they wouldn't change a thing.

The developers started with the assumption that the wireless application would look a lot like the Web application. Fortunately, they didn't stop there.

1.2 UI Design

The UI is the critical success factor in wireless applications, points out Bob Leo, Landstar's director of data management and administration, who participated in the initial project. "Wireless applications fail when not enough attention is paid to the user experience," says Leo. "It has to meet their needs and not get in their way. In B2B applications, the users want to get in, get their information, and get out. They're there for a specific purpose — not to browse."

Patrick Wise, Landstar's vice president of advanced technology, advises developers to look for every opportunity to minimize keystrokes by using menus, pick lists, shortcuts, and default options. For example, instead of requiring the BCO to key in a city name, the Landstar wireless application lets the user enter the

first few letters, and then it presents a pick list of city names that match. Typing in **MIA** prompts a pick list showing Miami, Miami Beach, and Miami Dade. **Lake** yields Lakeland and Lake City. Despite screen size limitations, the whole city name is displayed, wrapped if necessary.

Everyone who worked on Landstar's solution credits its success to the time invested in designing the application. "We brought in the owner-operators during our design session," says Wise. "That was critical."

In fact, the design team spent most of a week working with the BCOs. During that time, the Landstar developers showed the users storyboards of how the wireless application would work, received input about what the BCOs needed, and refined the development plan. "We started with many things," says Wise. "We funneled what we heard into what we could deliver and what would fit on a phone."

In addition to helping Landstar's wireless developers define which features were needed most and set feature priorities, the BCOs guided specific application behavior. For instance, one BCO pointed out that, as initially envisioned, he'd have to key in locations and load ID numbers. However, once he was logged in, the Landstar database already would know that information. Couldn't the developers present him with a list of his load numbers? They could. They did.

Once the developers realized that they could use existing database data to simplify the presentation on the phone, they expanded on it. The application uses drop-down boxes and other menu options to minimize data entry. Plus, because they knew what load the BCO had currently, they could insert the place from which the BCO was leaving as a default value in the driver location entry field.

User input wasn't limited to the initial design. The nine-month project included a three-month pilot phase, with 25 BCOs testing the system in the real world. To understand coverage issues, Landstar ensured that its testers came from two different networks. Here, too, the Landstar wireless developers incorporated feedback from the end users.

One change incorporated from the pilot phase was the addition of a user profile to the Web application. When they dial into the Web with a laptop computer, the BCOs can provide the values the software should use as their wireless defaults, such as the driver's truck type, capacity, and the type of loads for which he's qualified. There's no reason to frequently type in information that doesn't change. With the user profile, the BCO doesn't have to see extraneous information, so he has even less information to key in.

1.3 Programming

Most of the application is composed of active server pages (ASP), which are translated through the BlueMoon wireless gateway engine, and XML that interacts with the company's database. The BlueMoon engine, which is based on Java™ technology, provides the architectural infrastructure, scripting interfaces, and a device adapter library.

When the BCO makes a request, the BlueMoon engine interrogates the device that's asking the question and identifies the specific hardware, such as a Nokia Series 30, 40, 60, or 80 mobile device. The BlueMoon engine translates the WML of the truck driver's request into XML or ASP. In Landstar's case, it's XML. Landstar's own business logic uses XML to perform an ordinary database call on the company's IBM DB2 legacy databases, which run on IBM AS/400s.

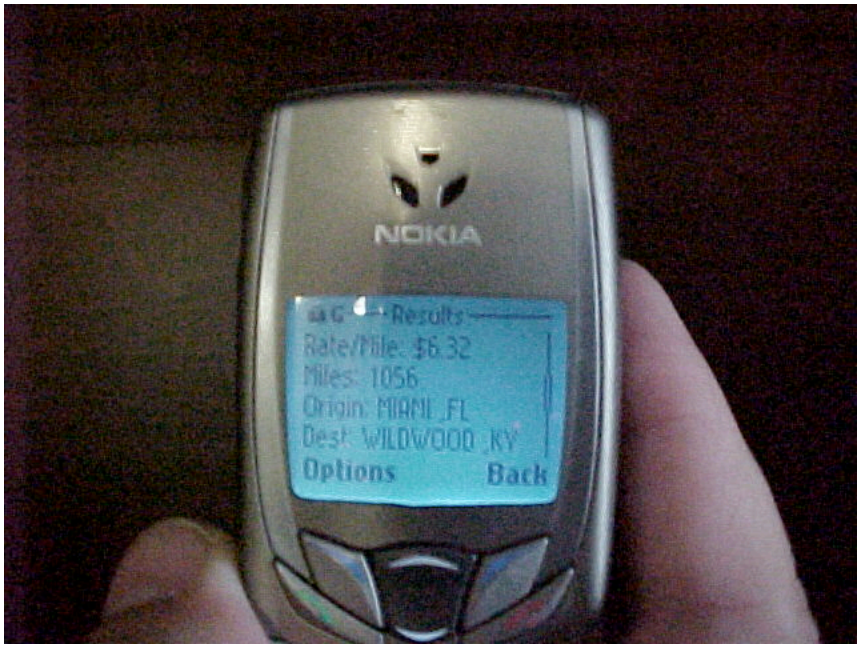

Figure 3: The device profile of the Nokia 6590 handset shown in this photo is stored in the BlueMoon engine, and is automatically retrieved when the BCO uses this device to log in.

The database's response is returned in XML and sent back to the BlueMoon engine. Because the BlueMoon engine retained the identification of the device when the request was made, it can use Extensible Stylesheet Language (XSL) style sheets like the one in the "XSL Stylesheet for nokiablueprint emulator Browsers" code sample shown in this document to format the information appropriately for the phone's display. Then the BlueMoon engine sends the screen to the BCO, who's had only enough time to take a sip of coffee while waiting.

The BlueMoon style sheets remove a huge UI burden from Landstar. BlueMoon's own databases store the specifications for each device, so the engine knows how many lines are on a given screen and the resolution of the phone. As a result, the XSL style sheet can send the information in 10-point type, so the application appears the same way on every screen.

In the application's n-tiered architecture, XSL Transformations (XSLT) customized for each device is the presentation layer, interacting with the business layer ASP pages and the data layer, XML. The Landstar developers don't have to write code in the Java programming language, C++, or another programming language. "It's all XML and ASP," says Leo.

The Landstar back end isn't WAP specific. In fact, it uses the same XML as the Web version of the application. The ability to reuse that code , Leo says, streamlined the company's development process immensely.

1.4 Trouble in River City: Solving Interface Problems

The development process wasn't trouble free. But the biggest problem that the programmers encountered wasn't wireless specific. "The interfaces to the legacy applications are where the bodies are buried," says Wise.

The legacy databases on the AS/400s stored the company's location data in a truncated format, which didn't necessarily resemble a city name. For example, is Phoenix *PHX* or *PHO*? Landstar couldn't expect truck drivers to make sense of the database's proprietary city codes, so the team constructed a translation table between the legacy database and the data sent to the wireless application.

That process took longer than initially expected and "a ton of effort," according to Leo. No one had grasped the size of the data set: something like 20,000 city and town names had to be translated and entered.

1.5 What's Coming?

The Landstar team is working to add speech recognition to its wireless application. To do so, they're repeating the process that made the current application so popular: the company is bringing in BCOs to test the speech application design first.

Adding speech isn't a technical challenge, says Leo. The BlueMoon engine can work with VoiceXML as easily as it does with WML. It's the design of a speech application that takes the time. "Speech completely changes the user experience," Leo says. "You have a totally different tree on the presentation layer."

At Landstar, color screens are still in the future. Their current application has no need for color because it's essentially a database update and retrieval requiring no snazzy graphics.

But Leo is waiting for increased bandwidth for wireless mobile devices. Then, he says, "providing maps and traffic routes will be feasible — and color will matter." Wise agrees. "2.5G and 3G will do wonders for our business."

2 Application Architecture

Figure 4 shows how the BlueMoon engine works its magic.

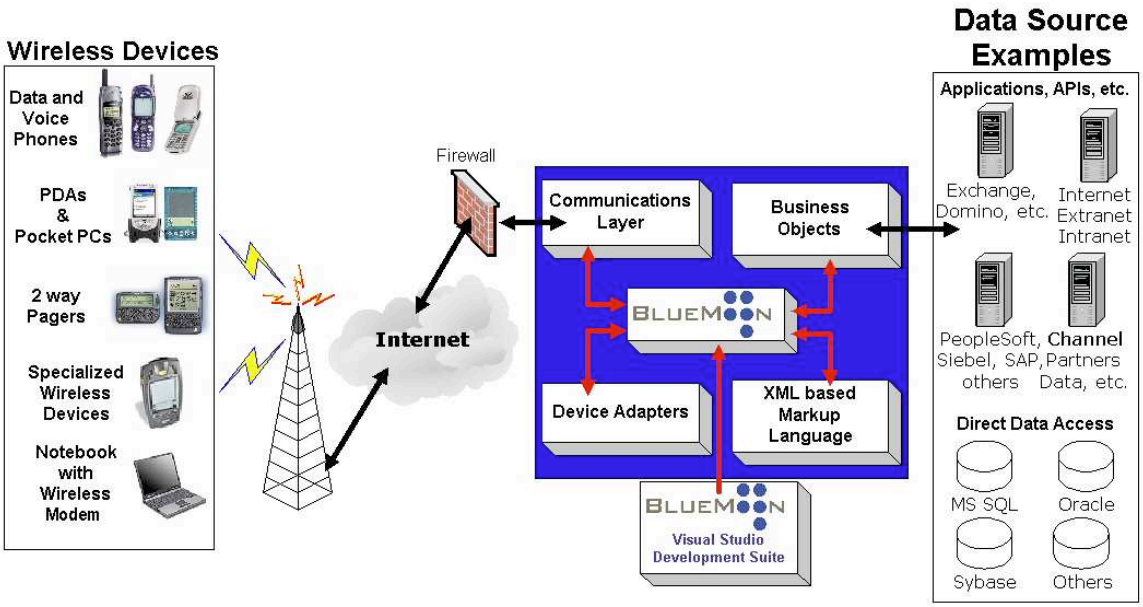

Figure 4: The BlueMoon architecture

The following sample code shows the Nokia style sheet embedded in BlueMoon. The style sheet interprets BlueMoon's version of WML, called Mobile Application Markup Language (MAML). It renders the content appropriately for the Nokia phone by translating the MAML "list," "symbol," and "br" tags. Most of the output is controlled by the attributes on the MAML list tag.

3 Code Samples

```
<?xml version="1.0" encoding="UTF-8" ?>
- <xsl:stylesheet id="nokiablueprint" version="1.0"
    xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:import href="wml_alt.xsl" />
    <xsl:strip-space elements="*" />
      - <!--
       ************XSL Stylesheet for nokiablueprint emulator Browsers***************
        up30.xsl              The device class base stylesheet

        Copyright (c) 2001 Phone Online
        Written by:                          Benji Harrell
        Details:

    -->
      - <!--
       ************************************
        Handle list tag here
         - attribute type="mi" will render as list as ul.
        ************************************
    -->
  - <xsl:template match="list">
      <xsl:value-of select="@title" />
      <br />
    - <xsl:choose>
      - <xsl:when test="@form='true'">
        - <select>
          - <xsl:if test="@display='combo'">
            - <xsl:attribute name="multiple">
                <xsl:text>true</xsl:text>
              </xsl:attribute>
            </xsl:if>
          - <xsl:if test="@title=true()">
            - <xsl:attribute name="title">
                <xsl:value-of select="@title" />
              </xsl:attribute>
            </xsl:if>
          - <xsl:if test="@name=true()">
            - <xsl:attribute name="name">
                <xsl:value-of select="@name" />
              </xsl:attribute>
            </xsl:if>
          - <xsl:if test="@tabindex=true()">
            - <xsl:attribute name="tabindex">
                <xsl:value-of select="@tabindex" />
              </xsl:attribute>
            </xsl:if>
          - <xsl:if test="@default=true()">
```

```
        - <xsl:attribute name="default">
            <xsl:value-of select="@default" />
          </xsl:attribute>
        </xsl:if>
      - <xsl:for-each select="li">
        - <option>
          - <xsl:if test="@value=true()">
            - <xsl:attribute name="value">
                <xsl:value-of select="@value" />
              </xsl:attribute>
              <xsl:apply-templates />
            </xsl:if>
          </option>
        </xsl:for-each>
      </select>
    </xsl:when>
  - <xsl:otherwise>
      - <!--
          ******************
          Render a simple list.
          ******************
      -->
    - <xsl:if test="@type='dl'">
      - <xsl:for-each select="li">
          <xsl:apply-templates />
          <br />
        </xsl:for-each>
      </xsl:if>
    - <xsl:if test="@type='ul'">
      - <xsl:for-each select="li">
          *
          <xsl:apply-templates />
          <br />
        </xsl:for-each>
      </xsl:if>
    - <xsl:if test="@type='ol'">
      - <xsl:for-each select="li">
          <xsl:number value="position()" format="1." level="single" />
          <xsl:apply-templates />
          <br />
        </xsl:for-each>
      </xsl:if>
    - <xsl:if test="@type='mi'">
      - <xsl:for-each select="li">
          <xsl:variable name="pos" select="position()" />
        - <xsl:for-each select="a">
          - <anchor>
              <xsl:value-of select="$pos" />
              <xsl:text>.</xsl:text>
```

```
                    <xsl:apply-templates />
                  - <go>
                    - <xsl:attribute name="href">
                        <xsl:value-of select="@href" />
                      </xsl:attribute>
                      <xsl:call-template name="process_vars" />
                    </go>
                  </anchor>
                </xsl:for-each>
                <br />
              </xsl:for-each>
            </xsl:if>
          </xsl:otherwise>
        </xsl:choose>
      </xsl:template>
        - <!--
          **********
          Handles the line break.
          The "clear" attribute is not supported in WML.
          **********
      -->
    - <xsl:template match="br">
        <br />
      </xsl:template>
        - <!--
          **********
          Handles rendering symbols.
          **********
      -->
    - <xsl:template match="symbol">
      - <xsl:if test="@value='$'">
          <xsl:text>$$</xsl:text>
        </xsl:if>
      - <xsl:if test="@value='%'">
          <xsl:text>%</xsl:text>
        </xsl:if>
      - <xsl:if test="@value='amp'">
          <xsl:text>&</xsl:text>
        </xsl:if>
      - <xsl:if test="@value='copy'">
          <xsl:text>(c)</xsl:text>
        </xsl:if>
      - <xsl:if test="@value='tm'">
          <xsl:text>(tm)</xsl:text>
        </xsl:if>
      - <xsl:if test="@value='r'">
          <xsl:text>(R)</xsl:text>
        </xsl:if>
      </xsl:template>
    </xsl:stylesheet>
```

4 Terms and Abbreviations

Abbreviation	Description
CRM	Customer relationship management
DB2	IBM's primary relational database product
MAML	Mobile Application Markup Language
PDA	Personal digital assistant

5 Resources

Bob Leo,
Director of Data Management and Administration
Landstar System, Inc.
(904) 390-1427
bleo@landstar.com

Build > Test > Sell

Developing and marketing mobile applications and services with Nokia

Get started

Use free resources available through www.forum.nokia.com: articles covering the latest technical and business issues, tools for developing and deploying applications and services, biweekly newsletters, and active discussion boards.

www.forum.nokia.com

Download tools and SDK's

Download free SDKs, emulators, simulators, and other tools that integrate with industry-leading integrated development environments.

www.forum.nokia.com/tools

Get specifications and documentation

Get comprehensive device and platform specifications, and find detailed technical documentation, tutorials, technical notes, case studies, FAQs, and more.

www.forum.nokia.com/devices
www.forum.nokia.com/documents

Get support and testing services

Access Nokia's wide range of technical support services, including case resolutions from our professional support staff and fee-based online support from our experts. Forum Nokia's Developer Hub services help with development and testing by providing both online and onsite access to servers, messaging centers, and the like.

www.forum.nokia.com /support

Take your applications to market

Take your applications and services to market through Nokia Tradepoint and Nokia Software Market. Other opportunities are available through Nokia Mobile Phones and other Series 60 licensees.

www.forum.nokia.com/business

Build your business with Nokia

Nokia works with selected leading developers in the games, branded media and content, and enterprise software industries. Submit your application to Nokia at www.forum.nokia.com/business.

www.forum.nokia.com/business