



SYSTRONIX



Mobile GPS Demonstration Platform

Project Description White Paper

Version 1.5 - 12/15/02

Summary: The purpose of the Mobile GPS Demonstration Platform (MGDP) project is to demonstrate a proof of concept for the integration of a number of emerging technologies including: Embedded/Realtime Java, Mobile, position-based applications, GPS (Global Position System) technologies, Wireless/Mobile connectivity, Web Services (XML, SOAP, WSDL, JAX-RPC, etc.), and server-side J2EE (including EJB and DBMS persistence). The MGDP has been constructed using commonly available, inexpensive hardware and software to illustrate that applications of this nature are very feasible using existing tools and technologies. It is hoped that the MGDP project inspires others to push the threshold of what can be done with position-based computing and the Java platform.





SYSTRONIX



Table of Contents

Introduction	3
<i>MGDP Topology</i>	4
MGDP Hardware	5
<i>MGDP Mobile Vehicle Hardware</i>	5
Systronix JStamp Controller	6
aJile Java Processor	7
<i>MGDP Web Service (J2EE Server) Hardware</i>	8
<i>MGDP Client Workstation Hardware</i>	8
MGDP Software	9
<i>MGDP Mobile Unit Software</i>	9
<i>MGDP Web Service (J2EE) Server Software</i>	9
<i>MGDP Client Mapping Application</i>	10
<i>MGDP Development Environment & Tools</i>	12
MGDP Future Directions	14
MGDP Project Sponsors	15
<i>aJile Systems</i>	15
<i>Systronix</i>	16
<i>Chaeron Corporation</i>	17



SYSTRONIX



Introduction

The Mobile GPS Demonstration Platform (MGDP) project demonstrates a proof of concept for the integration of a number of emerging technologies including: Embedded/Realtime Java, Mobile, position-based applications, GPS (Global Position System) technologies, Wireless/Mobile connectivity, Web Services (XML, SOAP, WSDL, JAX-RPC, etc.), and server-side J2EE (including EJB and DBMS persistence). The MGDP has been constructed using commonly available, inexpensive hardware and software to illustrate that applications of this nature are very feasible using existing tools and technologies.

The MGDP project consists of three major components:

- ◆ The MGDP Mobile Vehicle is built upon a Radio-Shack RC (Radio Controlled) truck platform, equipped with an Embedded Java Processor, GPS Receiver and Wireless Communication to the Internet which regularly transmits it's current PVT (GPS Position/VelocityTime) data using SOAP over HTTP to a back end Web Service over a wireless link to the Internet. The Mobile Vehicle also accepts remote commands to turn various functions on/off (siren/flasher/headlights).
- ◆ A MGDP Web Service (built on a J2EE server platform) that accepts real-time PVT information from the Mobile Vehicle and persists both the current and historical information. The Web Service also acts as a gateway that forwards commands to turn various Mobile Vehicle functions on/off (siren/flasher/headlights).
- ◆ A MGDP client application that displays the current and historical position information for the Mobile Vehicle on a moving map display by accessing the position data from the MGDP Web Service. MGDP client application can use static map images (eg. .jpg) or can access USGS Aerial Satellite map images of the continental USA from the TerraServer web service. The client application also exposes a UI that allows a remote user to turn various Mobile Vehicle functions on/off (siren/flasher/headlights). Two distinct client mapping applications have been built: a Swing-based standalone application, and a server-based Browser interface. The functionality of both these implementations are identical.

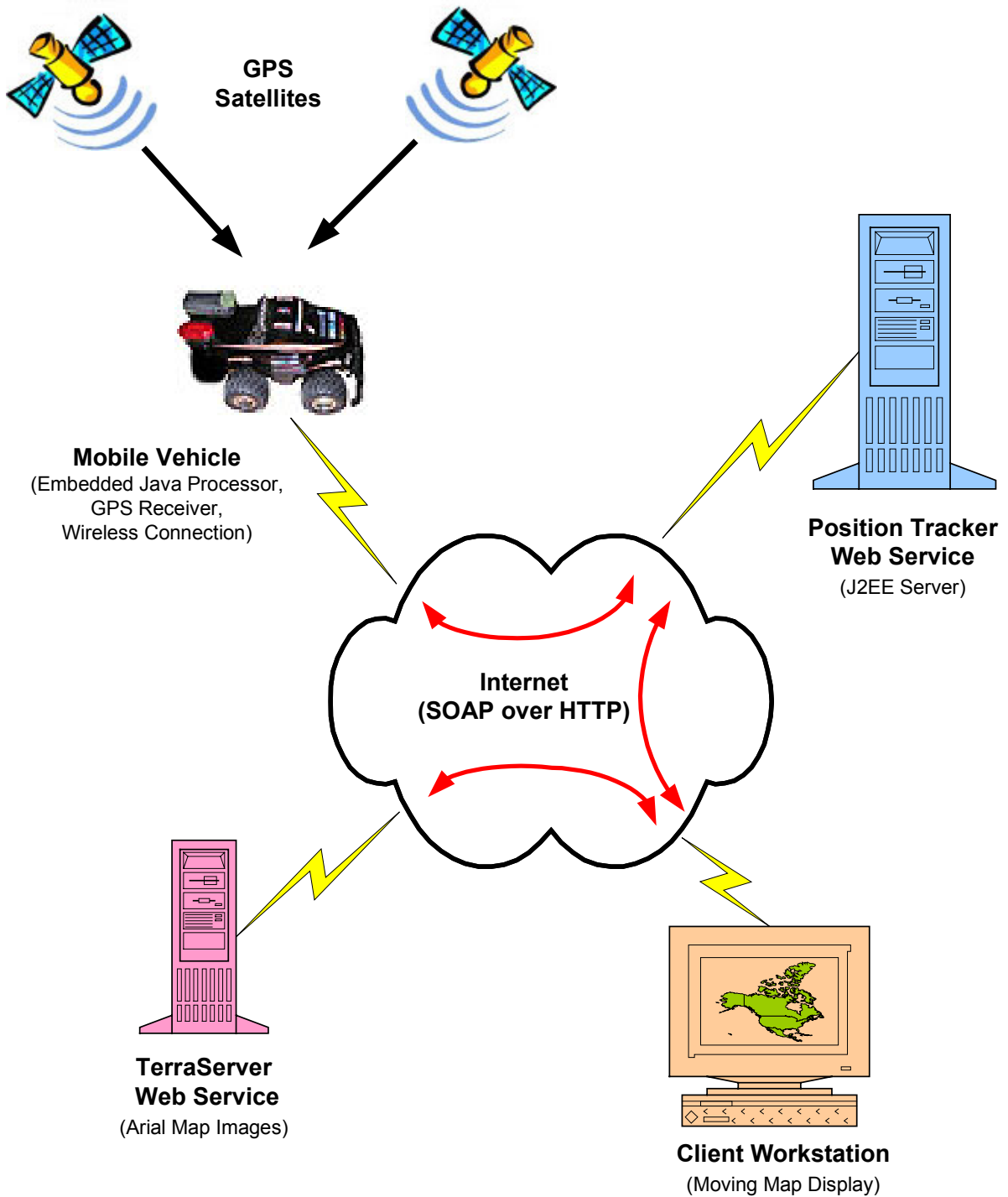
The following sections describe, in some detail, the topology, hardware and software components that have been employed in constructing the MGDP project.

It is hoped that the MGDP project inspires others to push the threshold of what can be done with Embedded/Real-Time Java processors, position-based computing and the Java platform.

Phase 2 of the MGDP project is in full swing and includes:

- ◆ Bluetooth enablement of the mobile platform (including cable replacement and remote config/control)
- ◆ Upgrading of the software infrastructure components to latest versions (JWSDP, JBoss, Tomcat, mySQL, etc.)

MGDP Topology



MGDP Hardware

MGDP Mobile Vehicle Hardware

The MGDP Mobile Vehicle incorporates the following hardware:

- ◆ Radio Shack RC (Radio Controlled) Truck
- ◆ Systronix JStamp Real-Time/Embedded Java Board.
- ◆ ajile Real-Time/Embedded aJ80 Java chip (on Jstamp board)
- ◆ Consumer GPS receiver with serial port interface (Garmin eTrex Summit)
- ◆ Consumer Wireless Connection device (Motorola GPRS mobile phone)
- ◆ Custom built hardware interfaces between JStamp and external devices (siren/flasher/headlights)
- ◆ Various custom cables
- ◆ Normal battery power is used for all devices.



MGDP Mobile Vehicle (GPS & Flasher visible at rear)

The MGDP Mobile Vehicle movement is controlled by an operator using the standard RC (Radio Controlled) joystick device.

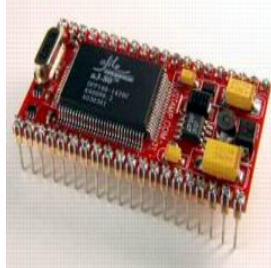
While in motion, the MGDP Mobile Vehicle JStamp controller reads obtains GPS PVT (Position/Velocity/Time) information from the GPS Receiver across a standard RS-232 serial port connection. It then transmits this information in real-time across a second serial port to the Internet across the wireless connection device to the back-end Server that persists this information. The back-end server returns indicators that tell the JStamp controller whether the siren, headlights or flasher should be turned on or off.



SYSTRONIX

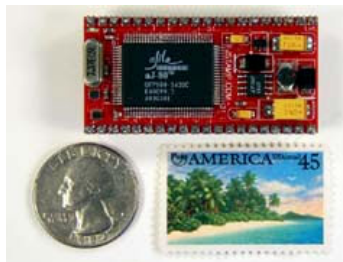


Systronix JStamp Controller



Closeup of Systronix JStamp controller with aJile Java Chip

JStamp uses the powerful new aJ-80 native Java processor from aJile Systems. This provides fast and efficient Java instruction execution, plus a small memory footprint - 2 to 3 times denser than code for 32-bit RISC machines. The aJ-80 includes a microprogrammed real-time Java thread manager so that a typical RTOS is not needed. It includes Java threading primitives implemented as atomic instructions, a priority-based preemptive scheduler, extremely fast context switching and interrupt response. The aJ-80 supports multiple application execution through "multiple JVMs". Applications execute in a deterministic, time-sliced schedule. Each application has its own thread management and event handlers.



JStamp is tiny!

JStamp is ready for serious work with 512 KBytes of SRAM and 512 KBytes of Flash (2 MBytes on JStamp+). Code can be executed from flash or SRAM. JStamp has five 24- mA I/O pins plus seventeen 8-mA pins (some of which also serve the UART and SPI functions). JStamp includes a SPI interface for easy peripheral and I/O expansion. The Jstamp development station adds hardware support for RS-232 serial I/O, and - very soon plug-on JSimm modules for Dallas 1-Wire network, CAN network, IrDA, RF, graphic LCD, analog I/O, power relays, and more. Of course, many current simmstick modules are JSimm compatible.

JStamp uses a JTAG interface for programming and debugging with aJile's JEM Builder and Charade development tools. JTAG pods are available from Systronix and Xilinx. Software requires Windows 98/NT/2000 and a PC parallel port.



SYSTRONIX



aJile Java Processor

aJile networked Java direct execution processors combine several innovative technologies to efficiently, safely, and robustly execute Java in wide variety of low-power embedded systems. The aJ-80 that is used in the JStamp is derived from the ground-breaking JEM™ Java processor that was first developed at Rockwell Collins. Incorporating direct execution of more than 99% of all Java bytecodes, and with an embedded thread manager that eliminates the need for an external real-time operating system (RTOS), the aJ-80 is the perfect Java processor for low-cost, low-power embedded applications.



aJile technologies include:

- ◆ **Direct JVM bytecode execution:** Software interpretation is a drag on performance. Java is the native instruction set for all of aJile's microprocessor and microprocessor cores. With more than 99% of Java bytecodes micro-programmed in hardware, aJile MPUs deliver unparalleled Java execution with a memory footprint that is 2 to 3 times more compact than similar 32-bit RISC applications.
- ◆ **Microprogrammed real-time Java thread manager:** aJile MPUs don't require an extra RTOS software layer. The on-chip real-time thread manager performs priority-based preemptive scheduling with extremely fast context switch times -- less than 1us.
- ◆ **Multiple application execution:** Embedded system designers enjoy the security and reliability of multiple JVMs when they develop aJile-based products. Discrete software applications are memory protected with selectable code & data sharing. Each application has its own thread manager and event handlers. And applications can implement unique memory management and garbage collection policies.
- ◆ **Custom instructions for application specific optimizations:** aJile's flexible architecture enables embedded system designers to optimize their Java applications. Using extended Java bytecodes and standard programming tools, designers can implement custom instructions that deliver 5X to 10X performance improvements.
- ◆ **Embedded programming tools:** System designers can use commercial Java IDEs to easily build and debug their embedded applications on aJile MPUs and MPU cores.
- ◆ **Integrated peripherals:** aJile MPUs and MPU IP cores include a complete portfolio of peripherals. Timers and interrupt controllers, memory and memory controllers, and a variety of communication interfaces are all integrated into the aJile architecture.
- ◆ **Integrated power management:** Mobile applications can go Java with aJile's comprehensive power management strategies, including sleep and standby modes that maximize battery life.



SYSTRONIX



MGDP Web Service (J2EE Server) Hardware

The MGDP Web Service is built upon the following hardware:

- ◆ 300 mhz Intel Pentium PC
- ◆ DSL connection to the internet

MGDP Client Workstation Hardware

The MGDP Client Workstation is built upon the following hardware:

- ◆ Intel Pentium PC
- ◆ Standard TCP/IP connection to the internet



MGDP Software

MGDP Mobile Unit Software

While in motion, a custom MGDP Java application uses the Chaeron Java GPS Access Library (a cross-platform GPS library that runs on almost any device supporting Java) to obtain GPS events that contain PVT (Position/Velocity/Time) data in real time. The application uses the kSoap library (a very small footprint web services library, designed for embedded devices) with a custom-written HTTP Transport handler to invoke a Position Tracking web service using SOAP across the internet (over a wireless link) to send it's PVT information (along with detailed GPS NMEA data) to the back end service. The SOAP response from the Position Tracking web service is parsed to retrieve indicators that determine whether the siren, headlights or flasher should be turned on or off.

The Mobile Unit uses the following software on the JStamp/aJile processor:

Software	Version	Function
aJile CLDC/MIDP Runtime	3.16	Core Java functions & HTTP network connectivity
Chaeron Java GPS Access Library	1.01	GPS interface and protocol parsing
Enhydra kSoap & kXML	1.0	Web Services/SOAP/XML Parsing over HTTP
Chaeron MGDP custom code	N/A	Overall demo application, glue and custom functions

MGDP Web Service (J2EE) Server Software

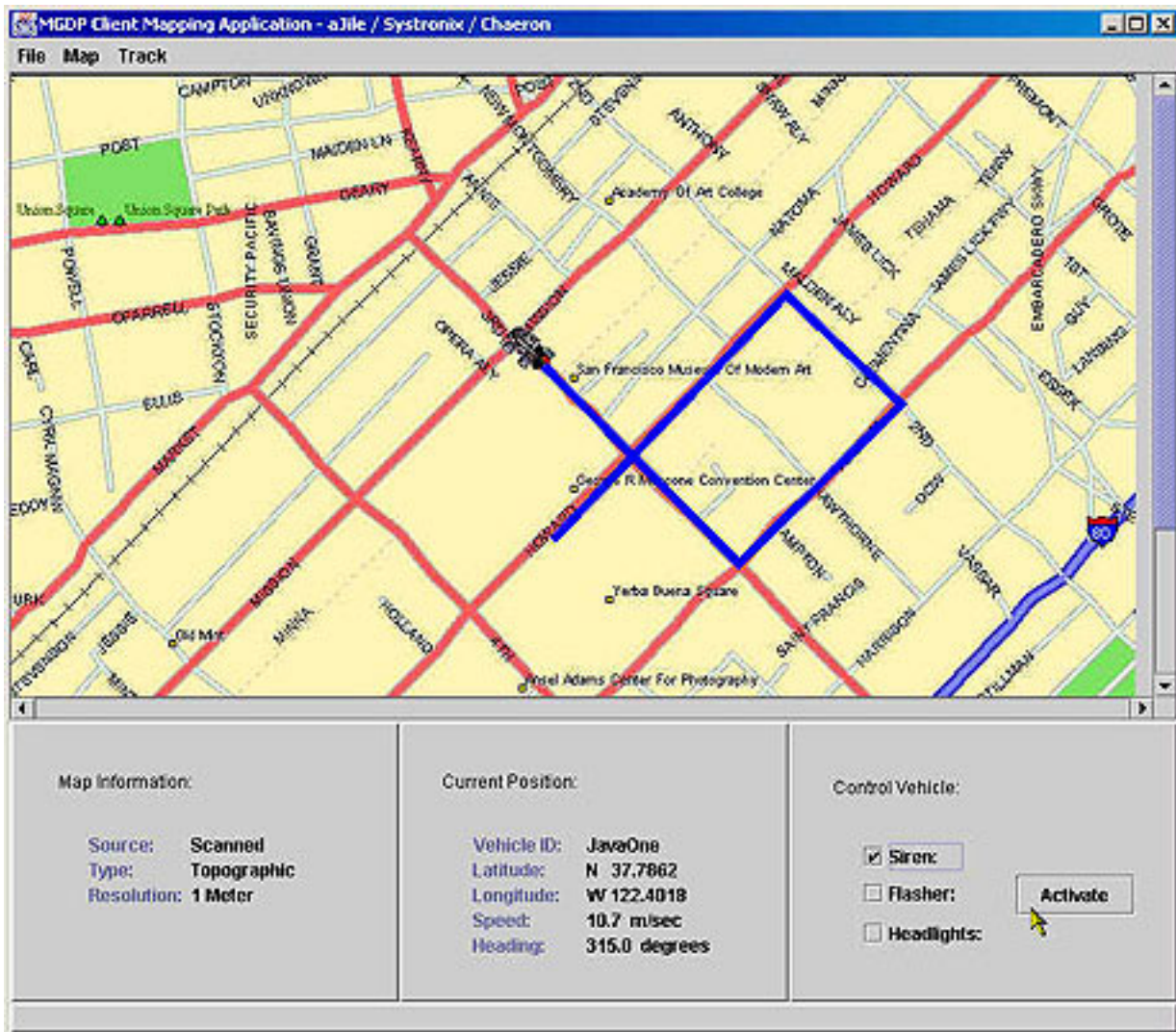
The back-end J2EE-based web service tracks and persists the Mobile Unit PVT (Position/Velocity/Time) information using a JAX-RPC SOAP interface running on the Apache Tomcat servlet engine. The data is persisted to a relational database using EJB components. Web Services interfaces are provided to retrieve the current Mobile Vehicle position information along with an interface that can be invoked to remotely control whether the Mobile Vehicle siren, headlights or flasher should be turned on or off. A WSDL definition is provided which defines the MGDP Web Services interfaces.

The MGDP Web Service (J2EE) Server Software uses the following software:

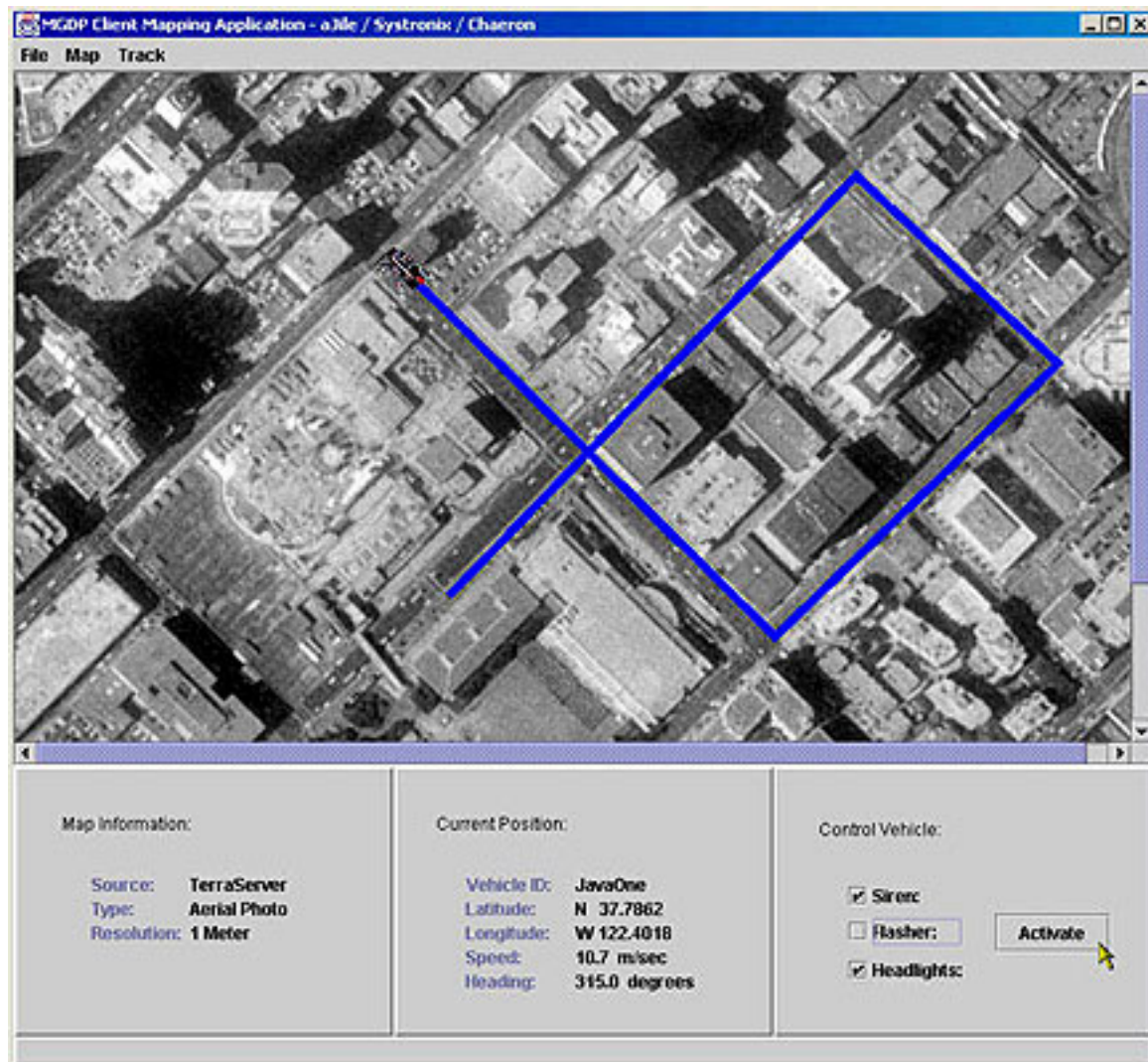
Software	Version	Function
JBoss	3.0.4	J2EE EJB Container
Apache Tomcat	4.1.12	J2EE Servlet Container
Sun Java Web Services Dev Pack	1.0_01	Web Services, SOAP, XML, JAX-RPC
Struts	1.0.2	JSP Presentation Layer Framework
Cocoon	2.0.1	XML/XSLT Presentation/Rendering Pipeline
Castor JDO	0.9.4.1	Java data binding framework
MySQL	4.0.5	Relational Database for persistent storage
Chaeron MGDP custom code	N/A	Overall server application and custom functions
Sun Java JDK	1.4 & 1.3.1	Java Runtime
Redhat Linux 7.2	2.4 Kernel	Server OS

MGDP Client Mapping Application

The MGDP Client Mapping Application uses a JAX-RPC SOAP interface to the MGDP Web Service to discover the current and historical position of the Mobile Vehicle. It then displays this information, using Swing or a Browser interface, on a moving map display. Images for the map display are obtained either from static scanned map images (.jpg) or can be obtained in real-time from the TerraServer Web Service (a publicly available web service implemented using Microsoft's .NET) which provides USGS Aerial Satellite maps for the whole of the continental USA. The MGDP client application also provides to select the source of the map (TerraServer or Scanned), the type of map (Aerial satellite photo, Topographic or Relief) and the map resolution (from 1 meter up to 16 km per pixel). The Client Mapping App also allows the remote user to turn on/off the Mobile Vehicle siren, headlights and/or flasher.



MGDP Client Mapping App: Displaying position on a static Topo street map (screenshot)






MGDP Client Mapping App: Displaying position on a TerraServer supplied Aerial Satellite view (screenshot)

The MGD Client Mapping Application uses the following software:

Software	Version	Function
Sun Java Web Services Dev Pack	1.0_01	Web Services, SOAP, XML, JAX-RPC
Chaeron MGD custom code	N/A	Overall client application and custom functions
Java Swing UI Library	1.4 & 1.3.1	Graphical, moving map display
Sun Java JDK	1.4 & 1.3.1	Java Runtime
TerraServer Web Service	N/A	USGS Satellite Aerial map images (Microsoft .NET)

MGDP Display - Microsoft Internet Explorer

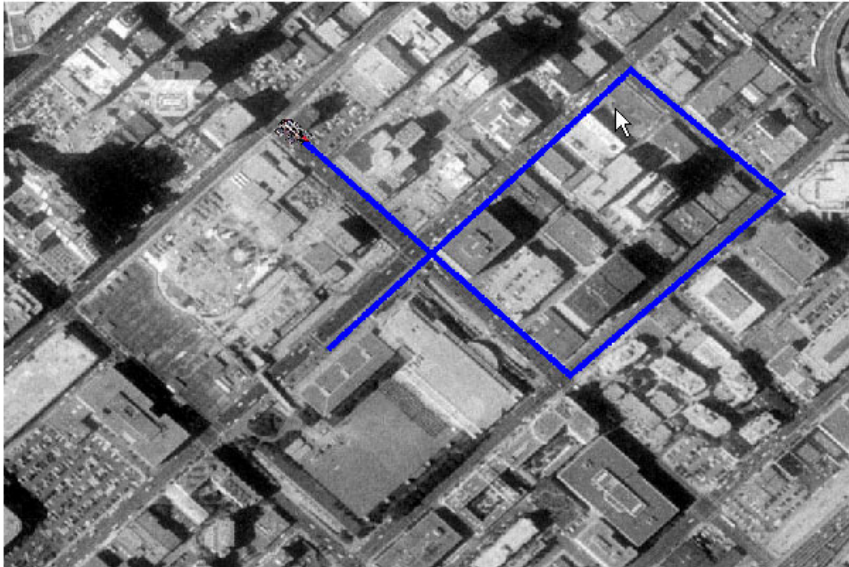
File Edit View Favorites Tools Help

[Back](#)   

MGDP Monitor

Controls

Refresh	None	Resolution	1 M	Type	Photo	Source	TerraService
Test Data	<input checked="" type="radio"/> Yes <input type="radio"/> No	Siren	<input type="radio"/> On <input checked="" type="radio"/> Off	Flasher	<input type="radio"/> On <input checked="" type="radio"/> Off	Headlights	<input type="radio"/> On <input checked="" type="radio"/> Off



Position/Velocity/Bearing

Longitude	-122.4018	Bearing	315.0
Latitude	37.7862	Velocity	0.0
Altitude	0.0	Time	0.0

NMEA Sentences

```
$GPRTE,1,1,c,*37
$GPRMC,V,,,,,050402.8,7,W,N*26
$GPRMB,V,,,,,A,N*13
$GPGGA,,,,,0,00,,M,,*68
$GPGSA,A,1,,,,,,*1E
```

Internet

MGDP Browser-based Mapping Client: Displaying position on a TerraServer supplied Aerial Satellite view (screenshot)



MGDP Development Environment & Tools

Various tools were used on all platforms (Mobile Vehicle, Web Service Server and Client Mapping Application) to develop, debug and test the MGDP software.

Some of the primary MGDP Development tools included:

Software	Version	Function
Sun JWSDP xrpcc tool	1.0_01	JAX_RPC, SOAP and WSDL Generation
Apache ANT	1.5.1	Build tool (XML-based makefiles)
JSWAT	2.11	Java Debugger
aJile JEMBuilder/Charade	3.16	Build/Run tools for the JStamp/aJile platform
JPadPro	4.5	Java source code editor
CapeClear NetTool	1.02	SOAP Message Monitoring/Transmission testing tool
Sun Java JDK	1.4 & 1.3.1	Java Runtime



SYSTRONIX



MGDP Future Directions

The MGDP project is intended to evolve and encompass other emerging and novel technologies to demonstrate the potential of the integration of technologies such as: Embedded/Realtime Java, Mobile, position-based applications, GPS (Global Position System) technologies, Wireless/Mobile connectivity, Web Services (XML, SOAP, WSDL, JAX-RPC, etc.), and server-side J2EE (including EJB and DBMS persistence).

Phase 2 of the MGDP project is in full swing and includes:

- ◆ Bluetooth enablement of the mobile platform (including cable replacement and remote config/control)
- ◆ Upgrading of the software infrastructure components to latest versions (JWSDP, JBoss, Tomcat, MySQL, etc.)

New functions and features will be added in the future. Some examples which are being actively considered for implementation in MGDP include, in no particular order:

- ◆ Integration of Embedded Java processor, GPS and Wireless devices into a smaller, more integrated form factor (eg. single board/single chip solutions)
- ◆ Onboard vision/imaging system which sends real-time images captured by the MGDP Mobile Vehicle to the back-end web service and client applications
- ◆ Voice transmission from/to the MGDP Mobile Vehicle.
- ◆ LCD Message display capability on the MGDP Mobile Vehicle
- ◆ Retrieval of messages posted in “mid-air” by the MGDP Mobile Vehicle.
- ◆ Position-enabled “guide” applications (eg. museum, art gallery, national monuments, etc.)
- ◆ Indoor position determination.
- ◆ Generic tracking Web Service for multiple mobile users.
- ◆ Storage and retrieval of position data using the GPSml markup language standard.
- ◆ JXTA support for position information dissemination.
- ◆ UDDI registry for position tracking web services.
- ◆ Security/Encryption support for position data
- ◆ Remote control of MGDP Mobile Vehicle movements from a client workstation application.
- ◆ Whatever else we think has a high “coolosity” appeal.
- ◆ Suggestions from the Embedded/Real-Time Java and Web Services communities.



SYSTRONIX



MGDP Project Sponsors

ajile Systems

The power of computing is moving beyond the desktop. The Internet is bringing a new generation of smart, distributed, networked applications to an incredible new array of consumer devices: smart phones, smart cards, PDAs, set-top boxes, information kiosks -- even gas pumps.

ajile Systems, Inc. is at the forefront of this evolution. Founded by a team of technologists who developed the world's first direct-execution processor for the Java™ platform, ajile develops and markets a family of microprocessors and IP cores that deliver high-performance, low-power and low-cost execution to embedded applications. ajile's innovative processors have been used by Mitsubishi Electric, and many other system developers who are leading the revolution.

ajile Systems was formed in July 1999 by a group of technologists from Rockwell Collins, Sun Microsystems, and Centaur Technologies. The principals of ajile Systems share a common belief that the rapid proliferation of Java technology, the Internet, and recent advances in CMOS process technology are extending the Java paradigm beyond the desktop and server. By providing a family of low-power networked single-chip Java microprocessors and IP cores, which includes a real-time kernel, ajile's founders are helping to bring the Java revolution to a wide variety of handheld, mobile and consumer appliances.

While still at Rockwell Collins, ajile's principals led a team responsible for the development of the JEM1, the world's first direct execution Java microprocessor, which achieved its first silicon in July, 1997. JEM technology enables low-power embedded applications to be programmed entirely and directly in Java and at the same time to achieve the efficiency and performance of comparable conventional embedded processors programmed in C.

ajile licensed Rockwell's JEM technology and is developing the ajile family of processors for low-power embedded networked applications.

Contact Information:

ajile Systems
920 Saratoga Ave, Suite 209
San Jose, CA
USA 95129

Phone: 408-557-0829
Fax: 408-557-8279

Email: info@ajile.com

Web: <http://www.ajile.com>



SYSTRONIX



Systronix

Systronix’s mission is to deliver embedded control development systems of exceptional value.

Systronix specializes in the 8051 family of 8-bit microcontrollers, and is particularly experienced with the Dallas 8051 supersets. In addition to the standard products you can see on the Systronix web site, Systronix has developed numerous commercially successful systems (and some not-so-successful) which are in use around the world today. Systronix has been in business since 1986 - long enough for many of our customers to have changed names more than once! Systronix is a privately held corporation.

Systronix manufactures and sells a number of products that incorporate the aJile Java chips, including:

Product	Description
JStamp	Revolutionary Real-Time Native Java Module. Tiny, low-power, low-cost, easy to use.
SaJe	Blazingly fast 32-bit real-time aJile controller executes Java natively!
SBX2	Plug-on support for LCD, keypad, 24 I/O bits, RS485/ RS232/IrDA.
8x1-Wire I/O Board	Plug on multiples of 8 digital I/O points to any 1-Wire network.
TILT & TILT.CAN	A low cost TINI sockets board complete with ethernet, 1- wire, RS232, power regulator and optional CAN interface.

Collectively, Systronix' employee's experience spans the development of the first commercial digital audio recording systems (at Soundstream), early optical recording technology development (at Digital Recording Corp), multi-microprocessor CPU design in 1982 (IBM, using 68000s), and numerous other technologies, most of which are now relics. We've created ASICs used in telecomm and high-end graphics systems.

Contact Information:

Systronix
555 South 300 East,
Salt Lake City, Utah,
USA 84111

Phone: 801-534-1017
Fax: 801-534-1019

Email: info@systronix.com

Web: <http://www.systronix.com>



SYSTRONIX



Chaeron Corporation

Chaeron Corporation provides software consulting services and was founded 10 years ago to assist corporations in the management, architecture, design, development and deployment of distributed applications and systems. Chaeron specializes in Web Services, Java, J2EE, EJB, Enterprise Messaging (JMS), GPS (position-based), B2B, EAI, and eCommerce systems.

Chaeron has extensive experience and a proven track record in leveraging the use of cross-platform, Web Services tools such as Sun's Java, J2EE (BEA Weblogic, Gemstone/J, WebSphere), Fiorano's Message Queue (FMQ), and Java Message Queue (JMQ), Blaze Software's Advisor and other tools in an object-oriented environment. Expertise with a variety of database management systems (Oracle, Sybase, etc.) is coupled with real-world use of technologies such as Java, C++, MOM (Message Oriented Middleware), Java Message Service (JMS), Enterprise Java Beans (EJB), Web Services (XML, WSDL, SOAP, UDDI) and CORBA to facilitate the development of robust business applications. Chaeron excels in utilizing these cutting edge technologies and tools in the delivery of mission-critical business applications and the implementation of B2B, EAI and eCommerce systems.

Chaeron Corporation also has extensive expertise in position-enabled applications. We offer many services to clients looking for location and GPS-enabled solutions, including consulting, design, custom development and more, on platforms that span consumer devices (eg. Handheld /PDA's, mobile phones), embedded Java and wireless platforms, desktops (PC/Mac/Linux in C/C++ or Java) and servers (unix, linux and mainframes).

Chaeron developed the open-source, Java-based GPS Access Library which was used as the core software foundation for the MGDGP, and provided custom software development for the project. Chaeron Corporation has also developed and continues to enhance the GPSSml markup language standard (XMLSchema) for GPS/position data encoding. The Chaeron GPS Java Access Library and GPSSml can be downloaded from the Chaeron web site.

Chaeron consists of a battle-hardened and experienced team of technology specialists (aka: the codestuds) that is intensely motivated and hyper-productive in achieving client goals. Chaeron's sterling track record is in part due to extensive experience in managing large, international, multi-location advanced technology projects, combined with comprehensive capabilities in Executive/Team/Project Management, Business Analysis, Architecture, Design, Development and Quality Assurance/Testing. Chaeron also brings strong executive management, inter-personal skills, successful development methodologies and industry expertise to the table to define user requirements and ensure that the final product is functional, on time and under budget.

Contact Information:

Chaeron Corporation
2040 Penetanguishen Road
Springwater, Ontario
Canada L4M 4Y8

Phone: 705-791-3996
Fax: 705-792-2043

Email: corp@chaeron.com

Web: <http://www.chaeron.com>