TRAMP
Traveling Repair and Maintenance Platform

Problem Statement

25 October 2001

1.The Problem

In 1923 Henry Ford introduced the assembly line to be able to build cars more efficiently. The assembly opened the era of mass production: You were able to get any car you wanted as long as it was black. A car is a complex system consisting of many parts. Maintaining such a system requires a very good knowledge of the functionality, structure and dynamic behavior of each of these parts and their interaction.

The 90s introduced the era of mass customization, the ability to mass produce products, but tailor them to individual people. An individual customer is now able to order a pink Mercedes, which will be produced on the assembly line.
Mass customization of consumer products leads to many challenging problems.

One problem is that each mass-customized system is now an individual product adapted for a specific user or purpose. The maintenance is now complicated, because the maintainer has to identify the particular system and its parts before they can start with the maintenance.

Combining both, complex systems and mass customization, introduces problems for the car mechanics: They need to learn a lot, the required knowledge for the repair processes for all the systems is now quite substantial. Furthermore, they are never up to date: the knowledge base is never constant, because new customized models with new variants are constantly produced.

The recent progress in computer hardware, networking technologies and software engineering has opened up an opportunity to deal with these problems.
Inheritance and design patterns allow us now to build software for customizable product-series with a very small range of serial numbers.

The ability to build smaller and smaller computers has led to light and energy-efficient wearable computers which can be worn on the body of the user for extended periods of time, in particular during the 8 hours of a typical mechanic‘s work day. Wireless networking has enabled us to access data at speeds of up to 10 Mbit/sec .

In the next 3 months you will be working on the TRAMP system. In this project we want you to investigate the use of wearable and mobile computers for the maintenance of cars. In TRAMP, we are moving away from the traditional desktop solution where the mobile mechanic moves to the computer to get the desired information. Instead, we want the information to move where ever the mechanic is currently located.

Many problems and challenges need to be addressed to implement this vision. In TRAMP we are looking for the development of new user interface metaphors allowing mobile workers to access information from a remote document database. We also want to investigate the possibility to support the mechanic when the network changes in terms of connectivy (e.g., Ethernet, WaveLAN, UMTS), available bandwidth, and quality of service.

2. Objectives

TRAMP's objectives can be divided in two parts. First, there are some overall goals that have to be kept in mind especially during the system design phase.

o Overall Goal: Develop a Car Maintenance System that supports the TRAMP scenario and provides as much capability as possible as constrained by the functional and non-functional requirements and the time assigned to the class

o Secondary Goal: Identify and evaluate new UMTS applications to support vehicle maintenance

Second, there exist another objective that focuses on the actual development process:

o Merge new system components with existing components from Inmedius and TUM

3. Scenarios

The overall goal of the TRAMP project is to demonstrate the technologies and concepts that will be developed by the class and which will provide capabilities that can be used as part of a visionary UMTS Mobile Maintenance system. In this section, the two different scenario types are given.

3.1 Visionary UMTS Scenario

Car owner Anton is driving on the Autobahn. His car breaks down with a problem that he cannot fix alone. With his UMTS phone, he calls the hotline of his car manufacturer, which notifies the closest workshop.

The workshop sends the mechanic Manfred to check Anton’s car. Some information and data which Manfred needs about the car are automatically sent by the car to Anton’s UMTS phone, which in turn sends them to the hotline and from there it reaches Manfred.

From the data Manfred concludes that the problem could be caused by the distributor. He therefore decides to get a new distributor from the spare parts room and puts into his repair-bag.

Manfred then determines the location of Anton's car and finds the shortest route to get there. After his arrival he checks the car and finds that there is indeed a problem with the distributor.

Replacing a distributor is quite a complex operation, and Manfred uses his access to UMTS to download the necessary data - drawings, cable labelings, and remove-and repair instructions - to do the replacement.

The sequence of instructions is shown on his head-mounted display.
After the repair is finished, Manfred sends the total time and parts used to the workshop which in return sends a bill to Manfred via UMTS.

Manfred prints out the bill and gives it to Anton, who then pays with his credit-card.

3.2 TRAMP Car Maintenance Scenario

John's head light turn signal does not work anymore, so he decides to go to a nearby garage.

When he arrives there, Toni, the customer representative at the reception enters the problem into his wearable computer, Spot. Toni wears a head mounted display and uses speech recognition and Inmedius' Wheel/Pointer to interact with his wearable computer.

Toni is advised by his wearable to reproduce the failure. Toni lets John sit in his car and activate the turn signal. It does not work.

Spot displays the following advice: "Let the customer drive the car to parking lot 235 where the customer should meet a mechanic." The customer drives to lot 235.

Meanwhile Brandon, a mechanic who is inside the garage, gets a notification (via wireless ethernet): "Show up at lot 235". Brandon also receives repair instructions as an IETM (interactive electronic technical manual).

Brandon puts necessary spare parts into his toolbox and goes to the parking-lot guided by navigation information displayed in his HMD. At the parking lot John is already waiting.

Brandon first checks the fuse-box following the steps automatically displayed inside his HMD. When he opens the fuse box - which is automatically detected by the optical tracker in his wearable system, the next instruction is displayed: "Check the fuse number 3123". The fuse is OK, so a new set of instructions starts to check the signal

Brandon then checks the lamp of the turn signal. He finds out that the lamp is blown, so he replaces it and checks whether the new one works (it does).

Brandon enters the payment information of John into his wearable (speech) and transmits the information via UMTS.

4 Requirements and Constraints

4.1 Functional Requirements

Wearable maintenance assistant. Wearable computer that assists the user (usually a mechanic) with nagivation and maintenance tasks.

Information retrieval. Retrieval and distribution of maintenance information from stationary servers to the wearable platform. This includes wireless transfer.

Navigation. The wearable computer can guide the user to the broken-down car.

Maintenance assistance. The wearable system assists the user with the maintenance of the car by presenting the appropriate repair documents at the right time.

Remote experts. The system should support communication with remote experts who can assist the mechanic with the maintenance tasks.

Synthetic experts. In the future (beyond the time of this class), the system should be able to support the user with synthetic experts and intelligent tutoring systems.

4.2. Nonfunctional Requirements

Multiple users. The system should support tasks that are performed by multiple users in concert, supplying each with the necessary information at the appropriate time.

Multimodal UI. The user interacts with the wearable system using different modalities, e.g. voice recognition and synthesis, three-dimensional maps on a head-mounted display, etc.

Reconfigurability. The user should be able to reconfigure his wearable system on the fly to accomodate different operating environments.

Hostile environments. The system should be able to operate in the presence of dirt, noise or other environmental difficulties.

Wearability. The wearable system should be designed so as not to encumber the user.

4.3 Constraints

Language . The system will be programmed in Java, except for performance-critical elements, which will be written in C++.

Framework. The system will be built in such a way that it uses components of the DWARF framework, and the components developed in the project should themselves be useable in other DWARF systems.

Design for Luther. The design of the system should enable it to interoperate with Inmedius' Luther architecture in future versions.

Repair Documents. The repair information for the head light turn signal will be authored using the AIMSS Interactive Electronic Technical Manual (IETM) authoring system

5. System Design

The overall TRAMP system architecture and components are defined in accompanying documents.

The TRAMP system will be built using the DWARF framework. This means that the various components will be useable within the framework. The components will be configured using XML files to support autombile maintenance.

6. Development Environment

At TUM room 3175 will be provided exclusively as a lab for the development of TRAMP.

The primary development hardware consists of Power Macintosch G3 computers running Mac OS X.
_The following tools will be used for the system development:

- o CVS for software configuration control
- o Lotus Notes for group communication
- o ANT as a build environment
- o JDK as a java programming environment
- o GNU C++, including cross-compilers for C++ development
- o REQuest for documenting requirements analysis
- o LaTeX or an XML-based system for documentation

7. Target Environment

The target environment for the wearable systems is Linux for StrongARM CPUs. This includes the Spot computers (based on availability) and (for testing purposes) Compaq's iPAQs. The target environment for stationary systems is Mac OS X on Power Macintosh computers.

8. Client Acceptance

A client acceptance test will take place on February 12 in room 3175, in the presence of a representative of Inmedius. On this date, the functionality of the system will be demonstrated.

9. Deliverables

- o A source guide for each subcomponent. There has to be a common design of this guide for all subcomponents. A proposal for the guide's structure is the Java package documentation. In addition to the information given by this proposal, an introduction for future developers of the software system has to be written.

- o A description of the subsystem interfaces and APIs. This will be the main content of the Object Design Document (ODD).

- o A suite of documents consisting of

  - Requirements Analysis Document (RAD)

  - System Design Document (SDD)

  - Software Project Management Plan (SPMP)

  - Object Design Document (ODD)

  - User Manual for the prototype system (this is to be a document containing the minimum procedures for operating the system after the project is complete)