TECHNISCHE UNIVERSITÄT DARMSTADT
Fachbereich Informatik
Fachgebiet Telekooperation
Prof. Dr. Max Mühlhäuser

SYECAAD: Ein System zur einfachen Erzeugung kontextsensitiver Applikationen

Rapid Development of Context Aware Applications

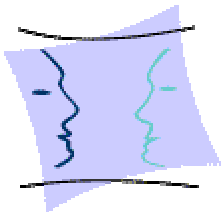
Diplomarbeit

Betreuer: Prof. Dr. Max Mühlhäuser

Erwin Aitenbichler

Jean Schütz

Darmstadt, 18.08.2005



Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 18.08.2005

Kurzdarstellung

Durch die zunehmende Verbreitung von Computersystemen und deren Vernetzung in unserer täglichen Umgebung sowie dem Preisverfall von Sensoren aller Art entstehen neue Interaktionsmöglichkeiten zwischen Mensch und Computer. Nach der Vision von Mark Weiser werden Computer zunehmend omnipräsent und können selbständig Funktionen ausführen, die individuell auf die jeweilige Situation angepasst sind und uns genau diejenigen Dienste zur Verfügung stellen, die wir im entsprechenden Augenblick und Ort benötigen. Die Programme solcher Systeme analysieren hierzu ständig Daten ihrer Umgebung, um in Abhängigkeit des Kontexts bewusst handeln zu können. Diese „intelligenten“ Anwendungen werden als kontextsensitive Applikationen *(englisch: Context-Aware Applications)* bezeichnet.

Zunächst zwei kleine: Sind in einem Büroraum Sensoren zur Erfassung der Personenidentität, der Lichtmenge die durch das Fenster eintritt und der Öffnung des Fensters an einem Rechner angeschlossen, so lässt sich eine Applikation realisieren, die automatisch das Licht im Raum einschaltet, sobald die Raumbeleuchtung durch das Tageslicht unzureichend wird und sich mindestens eine Person im Büro aufhält. Eine Vorschrift im Gebäude könnte lauten, dass alle Bürofenster geschlossen werden müssen, wenn ein Raum nach 18 Uhr verlassen wird. Steht nach 18 Uhr im Büroraum ein Fenster offen, kann eine Applikation automatisch denjenigen Mitarbeiter per SMS benachrichtigen, der als letzter den Raum verlassen hat.

Zur Erzeugung solcher Applikationen führt derzeit meist kein Weg an der expliziten Programmierung in einer entsprechend geeigneten Sprache vorbei. Somit ist die Erzeugung relativ aufwendig und kaum von Laien zu bewerkstelligen. Das in der vorliegenden Arbeit entwickelte Client-Server-System ermöglicht es, kontextsensitive Applikationen einfach, schnell und ohne Detailkenntnisse in Programmiersprachen zu erzeugen. Die Erzeugung erfolgt mit Hilfe eines intuitiven und graphisch-orientierten Modells. Applikationen werden dazu aus vorgegebenen Grundbausteinen aufgebaut, indem sie in geeigneter Weise miteinander verknüpft werden. Dabei werden Kontextdaten durch spezielle Sensorbausteine zur Verfügung gestellt. Applikationen und Bausteine können dynamisch während der Laufzeit hinzugefügt werden. Der Server verfügt über eine Laufzeitumgebung, auf der die Applikationen ausgeführt werden. Zudem werden Verwaltungsfunktionen bereitgestellt, die mittels des Clients aufgerufen werden können. Zwecks persistenter Speicherung und Austausch über Systemgrenzen hinweg werden die Applikationen in einem übersichtlichen XML-Format dargestellt, aus der das System jederzeit die Applikationen erzeugen und ausführen kann. Der Entwicklungs- und Zeitaufwand zur Erzeugung kontextsensitiver Applikationen wird durch das System deutlich reduziert und die Einstiegsschwelle zur Entwicklung gesenkt.

Abstract

New interactions between man and computer arise from the increasing growth of computer technology and its integration into our daily environment as well as from the price decline of all kinds of sensors. According to the vision of Mark Weiser computers are getting more and more ubiquitous and are able to carry out operations that are adapted to specific situations independently. Furthermore, they can provide us exactly those services we need at given time and place. The programs of such systems permanently analyze data of their surroundings to act consciously according to their context. These intelligent applications are called Context-Aware Applications. Two examples for context-aware applications are:

Consider a room in an office building in which some sensors are connected to a computer in order to collect the following environmental data: the identity of a person entering, the quantity of light getting in and the opening of a window. Now an application could be realized that turns on the light as soon as the room illumination provided by daylight is not sufficient enough and at least one person is staying in the room.

If a company regulation for a building demands that all windows have to be closed when leaving a room after 6 pm and if it happens that one window is left open, an application could send a message via SMS automatically to the person who has been the last to leave.

Presently, it is unlikely to generate such applications without having advanced programming skills. Thus, the generation of context-aware applications takes great effort and is hardly accomplished by a layperson. The client/server system that was developed in this work makes it possible to build context-aware applications quickly and without the necessity of advanced knowledge in programming languages. The generation is based on an intuitive and graphic oriented model. Applications are build using basic building blocks which are interconnected in an appropriate way. Thereby the context data is provided by specific sensor units. Both applications and units can be added dynamically at run time. The server provides a runtime environment in which the applications are executed and offers administrative functions to be invoked by the client. For the purpose of permanent storage and exchange over system boundaries applications are represented in a concise XML-format from which the system can generate and execute the applications at any time. The development effort and the expenditure of time to generate context-aware applications is clearly reduced by the system, thus the access barrier for development is highly decreased.

Danksagung

An dieser Stelle möchte ich mich bei einigen Personen bedanken: Vorerst danke ich *Erwin Aitenbichler* für die sehr gute und konstruktive Betreuung. Außerdem bedanke ich mich bei all jenen, die mir bei der Korrektur der Arbeit geholfen haben, insbesondere *Marc Dillmann*, *Manuel Hartl* und *Ayako Taniguchi*. Ein ganz besonderer Dank geht an meine Eltern, *Marie-Therese* und *Alain Schütz*, die mir das Studium ermöglicht haben.

Inhaltsverzeichnis