

CORD - Der Kontext-Management-Dienst

Ein Kontext-Management-Dienst ist eine Grundvoraussetzung für die kontextabhängige Unterstützung aller Aktivitäten zur Lieferung aktueller Informationen über die einzelnen Elemente einer Ambient Intelligence Umgebung wie Benutzer oder Geräte sowie zur Darstellung der physikalischen Umgebung. Während die Informationen über die einzelnen Elemente der Umgebung mit mehr oder weniger häufig wechselnden Werten beschrieben werden kann, beruht die Darstellung der physikalischen Umgebung im Allgemeinen auf aktuellen Sensordaten. Der vom Zentrum für Graphische Datenverarbeitung ZGDV e.V. entwickelte Kontext-Management-Dienst CORD wurde bereits in verschiedenen Projekten wie beispielsweise MAP und Mummy verwendet. Das Lösungskonzept liefert vor allem drei Instanztypen von CORD:

- Die persönlichen Instanzen, die den persönlichen Kontext wie Aufgaben, Benutzervorlieben, persönliche Kontakte und Zugehörigkeiten zu Organisationen verwalten,
- die Gemeinschafts-Instanzen, die Informationen über die entsprechende Gemeinschaft liefern, beispielsweise Organisationen und deren Beschäftigte, Abteilungen, Arbeitsgruppen, Gebäude oder Räume und
- die ortsabhängigen Instanzen, die an bestimmte Orte gebundene Informationen liefern.

Kennt die persönliche Instanz von CORD die organisatorischen Instanzen, zu denen der Benutzer Zugang hat, kann sie automatisch die ortsabhängige Instanz finden. Danach kann durch die Zusammenarbeit der beteiligten Instanzen ein umfassender Überblick über den Kontext gegeben werden.

Im Folgenden wird ein kurzer Überblick über die Herangehensweise von CORD gegeben. Im Anschluss daran erfolgt eine Beschreibung von BuddyAlert, einer Anwendung, die im Rahmen des MAP-Projektes entwickelt wurde und den CORD-Dienst verwendet.

Methode

CORD basiert vor allem auf den Konzepten des Semantik Web und der Semantik Technologie, mit der der virtuelle Bereich überschritten und bis in die physikalische Welt ausgedehnt wird. Sie stützt sich auf ein globales semantisches Informations- und Dienstenetz. Mit Hilfe der verwendeten Sprachen RDF und OWL werden sowohl Daten als auch Regeln zum Beurteilen dieser Daten und zum Austausch von Ontologien beschrieben. Diese Ontologien stellen das für das gemeinsame Verständnis der Kontext-Informationen benötigte Vokabular bereit. Abbildung 1 zeigt eine vereinfachte abstrakte Darstellung des CORD-Dienstes. CORD unterstützt als zentraler Dienst innerhalb einer Infrastruktur den Kontext-Bezug, der die zur Verfügung stehenden Informationen über den Benutzer-Kontext auf Grund gemeinsamer Schemata verwaltet. Außerdem gewährt CORD dem Nutzer einen (virtuell) zentralisierten Blick auf den Kontext und garantiert einen Persistenzgrad, der die Skalierbarkeit des Systems sowie die Unabhängigkeit und Asynchronität der Kontextinformation von Anbietern und Verbrauchern gewährleistet. Des Weiteren bietet CORD Mechanismen, die kontextbewußte Anwendungen von allgemeinen Unkosten, die beispielweise bei der Überwachung des Benutzer-Kontexts oder bei der Analyse interessanter Situationen anfallen, trennt.

CORD liefert unter anderem Ontologien zur Beschreibung von Terminal-, Ort-, Benutzer-, Aufgaben- und Dienstprofilen. Die Verteilung entsprechender Kontextdaten erfolgt durch eine Wiedergabekomponente. Der auf einer regelbasierten Situationserkennung basierende Trigger-Mechanismus erlaubt den Nutzern in relativ komplexen Situationen Mitteilungen zu abonnieren. Ein einzigartiges Merkmal des CM-Dienstes ist die Unterstützung konditionaler Werte. Dazu nutzt der Dienst einen Situationserkennungsmechanismus und ermöglicht, dass Kontextelemente alternative Werte für unterschiedliche Situationen annehmen können.

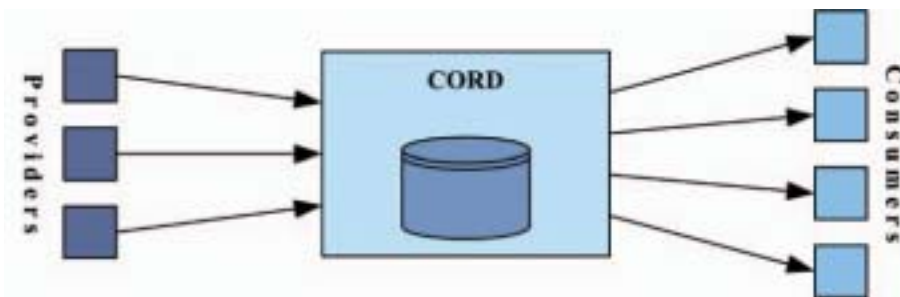


Abb.1: Eine stark vereinfachte abstrakte Darstellung von CORD.

Fig.1: A very simple abstract view on CORD.

Die BuddyAlert-Anwendung

Die Idee von BuddyAlert ist es, dem Benutzer durch eine Art Gedächtnisstütze zu assistieren. Damit demonstriert das System die Basisfunktionalität von CORD. Der Nutzer kann beispielsweise Aufgaben definieren, die mit Dokumenten und Personen verknüpft sind (den sogenannten 'buddies'). Trifft er auf eine der Personen, erinnert das System ihn an seine Aufgaben und bietet Zugang zu den entsprechenden Dokumenten. Die Dokumente werden mit Hilfe von CORD an die momentane Situation des Benutzers angepasst. Außer der Aufgabendefinition kann der Benutzer auch Memos aufsetzen, sie einem oder mehreren Buddies zuordnen und an sie verschicken. Abbildung 2 zeigt den Hauptbildschirm des BuddyAlert-Systems, den 'Alert Space'. Die Liste enthält alle Elemente, die für die zuletzt erkannten Situationen relevant sind: Aufgaben, ortsabhängige Memos und Mitteilungen zum gegenwärtigen Aufenthaltsort der Buddies.

Um Dokumente und Informationen, die gemäß ihrer Relevanz für eine bestimmte Situation gefiltert werden zu verwalten und zu pflegen benutzt BuddyAlert verschiedene Komponenten, von denen jede einen bestimmten Grad an semantischer Abstraktion darstellt (siehe Abbildung 3).

Der UI-Renderer stellt die Anwendungslogik des BuddyAlert-Systems in Form von dynamischen HTML-Seiten zur Verfügung. Diese HTML-Seiten werden mit Hilfe von vorgegebenen Templates erstellt und enthalten das Front-End der BuddyAlert-Anwendung. Der Situationsassistent kommuniziert mit und abonniert bei CORD. Er definiert Situationen in Hinblick auf die an CORD weitergegeben Regeln. Weitere Dokumente oder Informationen können ebenfalls mit diesen Regeln verbunden sein. Zusätzlich zu einer statischen Verknüpfung wurde auch ein dynamischer und automatischer Zugang entwickelt. Wird

eine besondere Situation entdeckt, das heißt, eine bestimmte Regel trifft zu, erhält die Situationsebene eine Benachrichtigung von CORD. Die Auswertung dieser Nachricht ermöglicht es der Situationsebene, die damit verbundenen Informationsobjekte pro-aktiv über die Render-Ebene auf das mobile Gerät des Benutzers zu übertragen.

Die Abfrageebene erlaubt dem Anwender sich mit verschiedenen Datenbanksystemen zu verbinden, die dann zur Speicherung von Aufgaben und Memos benutzt werden können. Es werden MySQL, Outlook und FastObjects Datenbanken unterstützt.

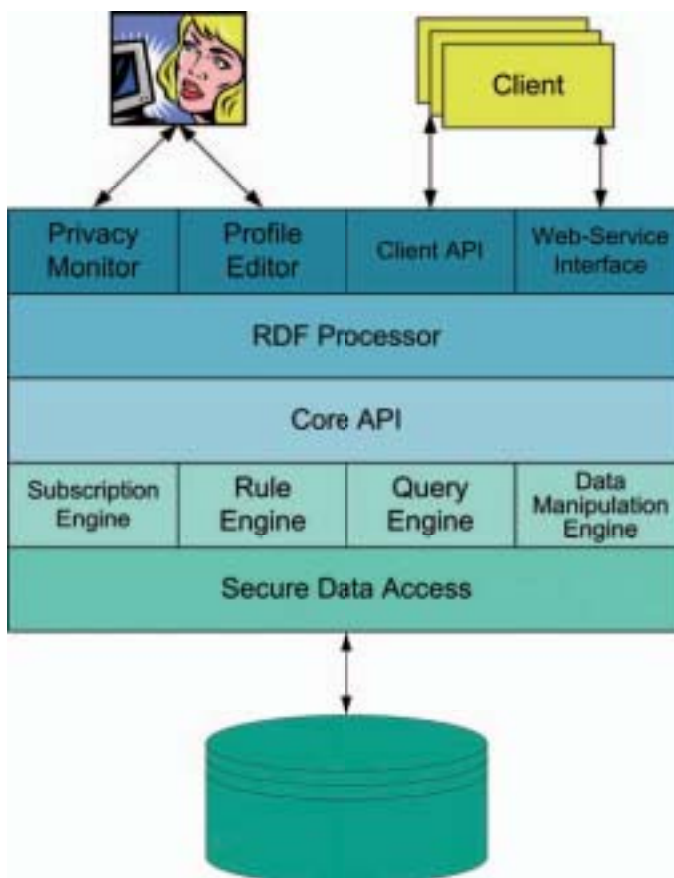


Abb.2: Die interne Architektur einer einfachen Instanz von CORD.
Fig.2: The internal architecture of a single instance of CORD.

CORD

- Kontext-Management-Dienst als Grundvoraussetzung zur Realisierung von Ambient Intelligence Umgebungen
- Drei Instanztypen – persönliche Instanz, Gemeinschafts-Instanz, ortsabhängige Instanz – die zusammenwirken und gemeinsam umfassenden Überblick über den Kontext geben
- Basiert auf Semantik Web und Semantik Technologie
- RFD und OWL als Sprachen
- BuddyAlert-Anwendung als Demonstration der Basisfunktionalitäten

Kontakt

Dr. Dirk Balfanz
Zentrum für Graphische Datenverarbeitung
ZGDV, Darmstadt
E-Mail: dirk.balfanz@zgdv.de

Saied Tazari
Zentrum für Graphische Datenverarbeitung
ZGDV, Darmstadt
E-Mail: saied.tazari@zgdv.de

CORD - The Context Management Service

A context management service is the main prerequisite for the context-aware support of activities, with the responsibility to deliver current information about the different entities within an Ambient Intelligence environment like users or devices as well as to represent the physical surrounding. The first group of information can be described with more or less infrequently changing values, in contrast to the representation of the physical surrounding, which is generally based on up-to-date sensory data.

The Computer Graphics Center ZGDV e.V. has developed a context management service, called CORD, which has been used in different projects, such as MAP and Mummy. The solution provides three main instance types of CORD:

- The personal instances that manage the personal context, such as tasks, user preferences, personal contacts, and organizational affiliations,
- the community instances that provide information in the context of the corresponding community, such as an organization and its employees, departments, work groups, buildings, rooms, etc., and
- finally, the location-bound instances that provide the context information bound to specific locations.

A personal instance of CORD has to know the organizational instances, to which the user has access and be able to automatically find the instance in charge of managing the context information bound to the user's current location. After that, an integrated view on context can be provided through the cooperation of the involved instances.

In the following, a short overview of the CORD approach is provided, followed by the description of BuddyAlert, an application developed within the project MAP using the CORD service.

Approach

The CORD approach is mainly based on the Semantic Web concepts and technology for breaking out of the virtual realm and extending into the physical world. It relies on a global semantic net of information and services, using RDF and OWL as languages that express both data and rules for reasoning about the data and for exchanging ontologies, which provide the vocabulary needed to reach a shared understanding of the contextual information. Figure 1 shows a simplified abstract view of the CORD service. CORD is the central service in an infrastructure, supporting context-awareness with the task to manage the available information on

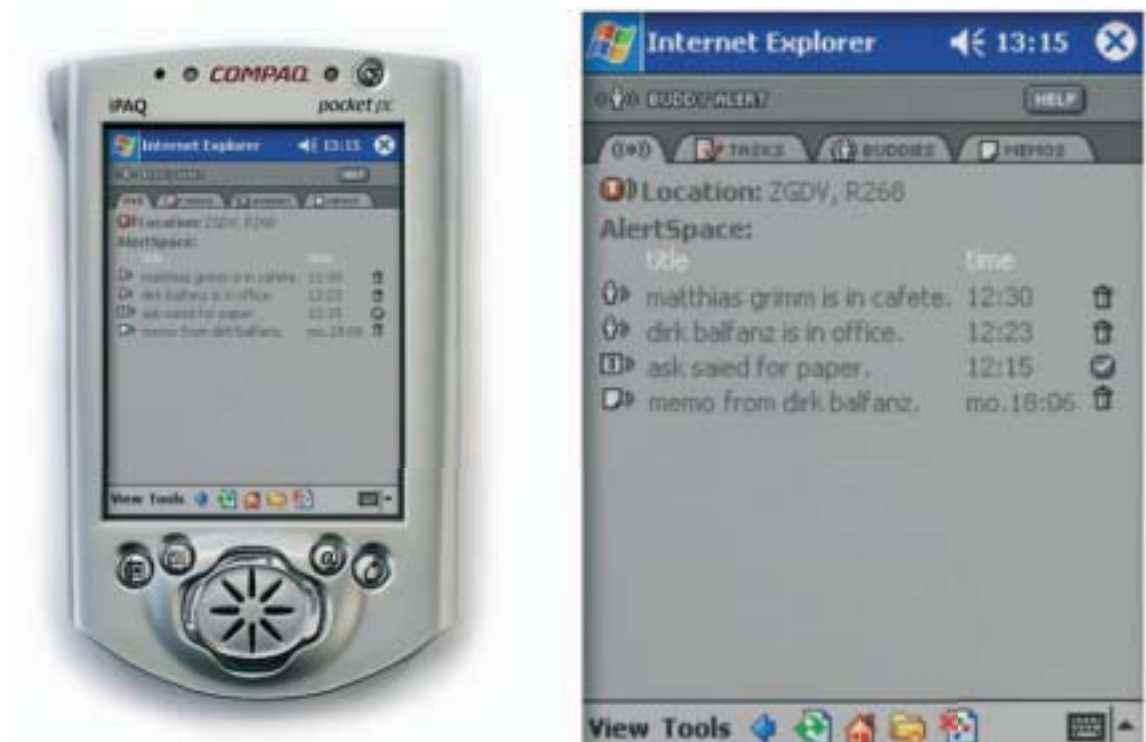


Abb.3: Eine Momentaufnahme des Alert Space.
Fig.3: A snapshot of the alert space.

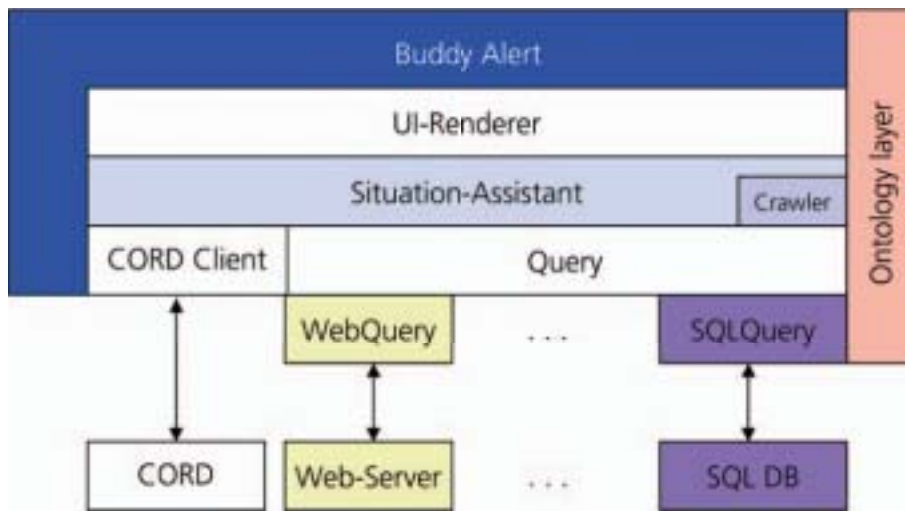


Abb.4: Die logische Architektur der BuddyAlert-Anwendung.
Fig.4: The logical architecture of the BuddyAlert application.

the user context based on shared schemas, to provide a (virtually) centralized view on it. Furthermore it guarantees the required persistence level, ensuring the scalability of the system and the independency and asynchrony of providers and consumers of the contextual info. In addition it offers shared mechanisms, relieving context-aware applications from certain common overheads like monitoring the user context and recognizing interesting situations.

CORD provides ontologies for describing terminal, location, user, task, and service profiles, just to name a few. The distribution of related context data is handled using a replication component. A triggering mechanism, based on the rule-based situation recognition engine enables consumers to subscribe for notifications in rather complex situations. A unique feature of the CM service is the support for conditional values that uses the situation recognition engine and allows context elements to have alternative values for different situations.

The BuddyAlert application

BuddyAlert follows the key idea of supporting the user with a kind of memory aid and demonstrates the basic functionality in this regard. The user may, for instance, define tasks linked with documents and persons (so called 'buddies'). Meeting one of the addressed persons would make the system remind the user of the tasks, offering, in addition, access to related documents. The

documents are adapted to the user's current situation, which is gained from CORD. In addition to defining tasks, the user can compose memos, address them to one or more buddies, and post them to a location. Figure 2 shows the main screen of the BuddyAlert system, the 'Alert Space'. The list includes all items that are relevant to the latest recognized situations: tasks, location-sensitive memos, and notifications concerning the locations of buddies. To manage and maintain documents and information that are filtered according to their relevance for a specific situation, BuddyAlert uses different components, each of which represents a specific level of semantic abstraction (see figure 3).

The UI-Renderer makes the application logic of the BuddyAlert system available in terms of dynamic HTML pages that are generated from given templates and comprise the BuddyAlert application's front-end. The situation assistant communicates with and subscribes to CORD. It is responsible for defining situations in terms of rules that are passed to CORD. Additional documents or information may be linked to those rules as well. In addition to static linking, we have also developed a dynamic and automatic approach.

When a specific situation is detected, for example when a rule comes true, the situation layer receives a notification from CORD. Evaluating this message enables the situation layer to push the connected information object(s) proactively to the user's mobile device via the render layer. The query layer enables the user to connect to different database systems, which can be used for storing tasks and memos. The databases supported are MySQL, Outlook, and FastObjects.

CORD

- Basic context-management services for the realization of Ambient Intelligence
- Three main instance types of CORD – the personal instance, the community instance, the location-bound instance – which provide a central view on the context information
- Based on Web semantic
- Using RDF and OWL as languages
- BuddyAlert-application demonstrates basic functions

Contact

Dr. Dirk Balfanz
Computer Graphics Center ZGDV,
Darmstadt, Germany
E-mail: dirk.balfanz@zgdv.de

Saied Tazari
Zentrum für Graphische Datenverarbeitung
ZGDV, Darmstadt
E-Mail: saied.tazari@zgdv.de