Towards a Context Aware Mobile Community Application Platform

Christian Menkens
Center for Digital Technology and Management
Technische Universitaet Muenchen
Munich, Bavaria 80290
Email: menkens@cdtm.de

***Abstract*—On the Internet, all sorts of community applications, especially social community applications, gain more and more importance every day. People in every type of community are eager to communicate, collaborate and inform themselves and others about news and information. So far the user interfaces for community applications have been Internet connected computers or advanced smart phones running standard web browsers. Although, todays mobile technologies and networks offer large sets of context information about mobile users, none of the existing mobile extensions of existing community applications tries to leverage this valuable data. This paper presents an approach to design and architect an IP Multimedia Subsystem (IMS) based Context Aware Mobile Community Application Platform (CAMCAP) that consolidates, federates and processes the most important context data of mobile community users before it provides it to community applications and clients. Furthermore, as proof of concept, this work presents the design and implementation of a University campus community application prototype based on the proposed CAMCAP.**

***Index Terms*—Context, IMS, Presence, Community, Mobile.**

I. Introduction

Community enabled applications on the Internet grow at a high rate and especially social community services such as Facebook and Myspace are leading in this field of Internet applications and services. Till now, the primarily used interface devices for these services and applications have been Internet connected personal computers with up-to-date web browsers. In the last two to three years smart phone devices appeared on the consumer electronics market. Now they are capable of accessing state of the art web applications, so they enable users to use their favorite community applications on the go. During that time, also community application developers recognized the trend that users wanted to use their community applications while on the move, so most large community applications and services developed some sorts of mobile extensions for their applications and services.

A. Problem

Unfortunately non of these current social community application extensions, mentioned above, do actually tab into the huge amount of context data and information that would be provided by community users that move and interact constantly in the real world and their virtual community using high end mobile devices or smart phones. The question is, how can a platform consolidate and federate all this context information and provide it to community applications?

B. Contribution

The aim of this research work is to propose the design and architecture of a Context Aware Mobile Community Application Platform (CAMCAP) that provides all means necessary to allow developers to implement comprehensive context aware mobile community applications. Using the CAMCAP as a platform, these community applications connect users within a community, let them communicate and collaborate and have constant access to the most up-to-date and exhaustive mobile user context data that is available. In addition to that, as proof of concept, this work outlines the implementation of a CAMCAP prototype including a University campus social community application running on top of it.

II. Background

A. Context

Context is any information that can be used to characterise the situation of entities [2]. In a mobile community the set of entities consists of mobile community users that use their mobile device to access a community application in order to communicate and interact. The context of a mobile community user is defined as all the directly user related data that can be accessed at a defined point in time. In this research work, context is not perceived as a matter of user location [2] but as part of a process the mobile user is involved in [3]. So a mobile user context includes all data that can be determined at the current point in time as well as all historical context data that is stored in the context data storage of the platform.

B. All-IP Networks and IP Multimedia Subsystem (IMS)

In the move towards a converged All-IP network architecture the IP Multimedia Subsystem (IMS) [10] standard defines a generic Next Generation Networking (NGN) architecture. It is a set of specifications for offering mobile and fixed Voice over IP (VoIP) and multimedia services. The IMS standard was introduced by the Third Generation Partnership Project (3GPP) as a part of their standard in Release 5. The standard supports multiple access types, including GSM, GPRS, EDGE, WCDMA, UMTS, Wireline broadband access and WLAN. IMS truly merges the Internet with the cellular

and fixed telecom world and uses cellular technologies to provide ubiquitous access as well as Internet technologies to provide appealing services. It uses the Internet Protocol (IP) [5] as its underlying network protocol and the Session Initiation Protocol (SIP) [6] as signaling protocol. In addition, it enables the integration of enabling services, such as presence [9], location [8], messaging [7], video, voice, picture and text, into one application, supports charging and Quality of Service (QoS) as well as deployment and re-use of applications and services.

III. Efficient Community Management

A. The Notion of Geographically bounded Communities

The social graph of a user grows steadily and studies [4] show that most people manage more than 50 contacts in their social community applications and services, some even more than 100 or 200 contacts and that the average is at about 144 contacts with a large median of 180 contacts. Nevertheless most people do not interact, communicate and collaborate with all of their contacts regularly and typically contacts within the same geographical region are the most frequented communication, collaboration and interaction partners.

So, in an effort to build a more scalable and manageable contact list / group list feature within the CAMCAP it is necessary to group a user's contacts into different, by geographical properties defined, contact lists. These properties, define the geographical area in terms of location, size and definition. This grouping of contacts, leads to the fact that a user is a *member* of many different communities that are all defined by geographical parameters but is *actively assigned* to only one of these communities at a time. The community he/she is assigned to is the community he/she will primarily communicate, collaborate and interact with. For occasional communication, collaboration and interaction needs with contacts from other geographically bounded communities the user can switch his/her personal community view to, and so interact with, any community he/she is member in.

B. Community Sets

Typically, social communities are not flat structures but contain some internal structure and organization. In addition, users are not just members of one global social community but of smaller communities that represent parts or fractions of the global community. In order to manage large communities more efficiently and to help users to manage their communities in a more user friendly way CAMCAP introduces community sets. Community sets are based on "Set Theory" and define sets and subsets with no limitation of how many levels of subsets there are. Typically a user is member or several sets such as geographically bounded communities and several subsets within these sets. Figure 1 show a graphical representation of a set with two subsets.

This user is part of one main social community set called "University" that is divided into three subsets. Two of them represent sports clubs the user is member of, "Basketball" and "Soccer" and the third one a class he/she is registered for and taking: "Class". The class subset is divided into subsets again and the user is member of the project group, "Project".

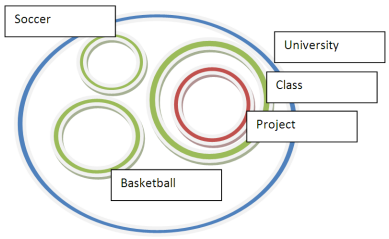

Fig. 1. University Campus Community Sets

IV. Platform Design

This section presents details about what kinds of context data and information the CAMCAP consolidates and how this context data will be processed and provided. In addition to that it discusses the idea of working with geographically bounded as well as sets of communities to support more complex community structures and enable easier handling of these structures.

A. Consolidating Context

As mentioned in section II-A this work focuses on consolidating context data that is directly related to mobile community users. This section presents and discusses some of the, for mobile community applications, necessary, important and beneficial context data entities including the sources they can be consolidated from and different update mechanisms and possibilities the CAMCAP needs to manage.

- **Location** - Location is one of the most essential context information entities for mobile applications. At the moment there are many different types of location data providing technologies like GPS, Cell ID, Wifi, Bluetooth, etc. available and they differ heavily in the way location data is determined and provided.
  Source: First party location systems such as GPS, Wifi and Bluetooth determine the location at the end user device, third party location systems such as Carrier supported Cell ID determine a mobile device location within the network and do not need any active support from the client application or mobile device.
  Update: At first party location systems the client needs to actively publish the location information to the CAMCAP, whereas at third party location systems the system can request the location information from the core network and update it directly within the CAMCAP.
- **Presence** - In order to be informed about another users current personal status, his/her ability to communicate, his/her willingness to communicate and collaborate and other interaction parameters it is essential to have exhaustive, up-to-date and accurate presence information. Presence data can be provided by various different sources and so can be divided into four different types of presence data, Device Presence, Network Presence, Service Presence and Personal Presence.

– **Device Presence** - Device Presence maps primarily to the capabilities of the device(s) the user is using to interact with the community. This includes for example screen resolution parameters, processing power, graphical representation parameters, audio parameters, user input and output parameters and more. One can imagine that these parameters change drastically if a user moves from his/her mobile phone to a high end desktop computer.
  Source: Primarily the client itself. For popular and known clients this information could also be provided by device profiles stored within the platform.
  Update: Can be done automatically by tracing all user/device/client registrations and requesting Device Presence data from the client or the client profile storage.
– **Network Presence** - Network Presence represents network parameters for all devices a user is logged on with. This includes for example bandwidth, delay, access technologies, network location/addressing, etc.
  Source: Primarily the network core and client access networks. For advanced parameters, the device could assist the core and access network, and actively report detailed parameters to the platform (e.g. all access points in radio reach, etc).
  Update: Can be done automatically by monitoring network access and data transmission parameters of all clients a user is active on the platform/network with.
– **Service Presence** - Service Presence represents a higher level (application layer) view on presence and includes information about all services a user is registered for or subscribed to.
  Source: The users service contracts and subscriptions with service providers such as telecoms and Internet based service providers as well as, for dynamic presence data, the users devices where different service clients and applications are installed on.
  Update: If all presence data providing sources are registered and accessible at the CAMCAP , all updates and presence data management can be done automatically.
– **Personal Presence** - Personal Presence represents all information known from Instant Messaging systems like ICQ, Skype, MSN etc and Social Network Applications like Facebook and MySpace. It defines the willingness and ability of a user to interact, communicate or collaborate using any of his/her devices or services. Often, this includes internal and external parameters like mood, business schedule/meetings, personal schedule etc and examples for Personal Presence status information are Online, Offline, Away, Busy, Open for Chat, Not Available, Bored, Relaxed, and many more. This presence information is unspecified and many applications defined their own sets of personal presence status options, so it can't be fixed and needs to be open and prepared for extension in many ways.
  Source: All services a user is registered or subscribed to and for which the CAMCAP has a Personal Presence Data Aggregation (see IV-B) profiles and interface for.
  Update: The user needs to manually set his/her Personal Presence information/data within a service client/application that is connected to the CAMCAP. In advanced scenarios and after the CAMCAP successfully learned and inferred some regularities (see section IV-B) for presence parameters of a user, the platform could set and configure some Personal Presence data automatically.

- **User Type/Community Affiliations** - This information represents details about a users role(s) as well as the community/communities a user is registered for. Examples for that would be a University Campus community where users can be in rolls such as lecturer, professor, student, visitor etc.
- **Date/Time** - Represents date and time information for all users and manages time zones within the CAMCAP. In addition to that it provides support for managing a users CAMCAP context data history and provides means for snapshots as well as continuous context data storage.
- **Calendar and Tasks** - Represents information about a users personal or business schedule which including all meetings and tasks. The CAMCAP manages, combines and processes this information and distributes it out to all clients a user uses.
- **External (social) community services** - This information represents links to external (social) community services and the users social connections that exist there. The CAMCAP manages and combines all internal and external social connections and relations in order to build a consistent social graph for a user.
- **External Internet services and applications** - This information represents links to arbitrary external internet based services that provide additional context information. Examples for these services could be video portals such as YouTube, VideoLoad etc or shopping and travel portals such as Amazon or Expedia.

B. Managing and Processing Context

This section presents an overview of the context management system within the proposed CAMCAP. Figure 2 shows a graphical representation of the architecture of the system.

The context management system consists of the following components.

- **Data Aggregation Layer** - This layer realizes a plugin and data profiles based concept for context data sources which allows an easy integration and adaptation of different context data sources and context data providing services. The profiles represent an abstraction from the network, service and protocol details and the plugin concept

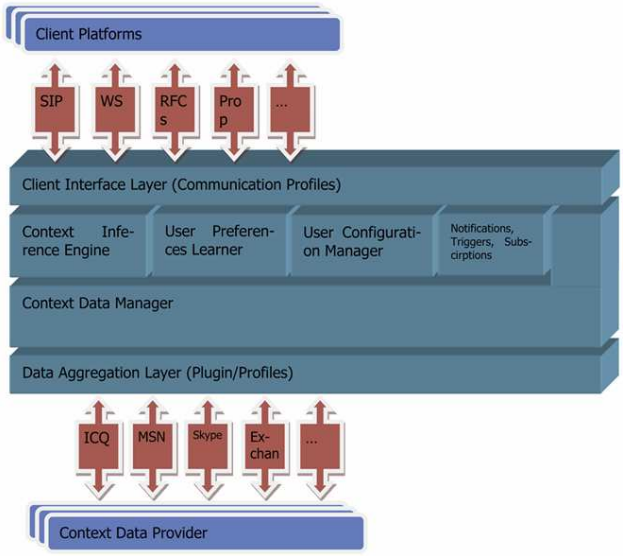

Fig. 2. Context Management and Processing Architecture

lets administrators add and remove context data providers at run-time. Examples for context data providers with very different protocols and service parameters are Instant Messaging systems such as ICQ, Skype, MSN, AIM etc or business management systems for calendar and tasks such as Microsoft Exchange, Outlook, Lotus Notes, etc. In order to import data from external context data providing services (e.g. calendar and task management, social community, ...) users need to register their services with the CAMCAP and if necessary provide user credentials in order to allow CAMCAP to import context data. These credentials are stored in the secure User Configuration Manager (see below) and provided to this layer upon request.

- **Context Data Manager** - This component realizes the data storage for all raw context data and information. It realizes an extendable and with semantic information enriched abstraction from the details of the different context data sources. Furthermore, the Data Manager consolidates the data provided by these sources into one defined set of context data. Extendability is provided by the possibility of plugging in new data sources at any time and semantic information such as categories describe context data in more detail.
- **Context Inference Engine** - This component uses inference algorithms to combine raw context data elements to higher level context information.
- **User Preference Learner** - This component uses raw and higher level context information to find regularities in a user's behavior and preferences. It uses these regularities combined with user defined preferences for services and application for assisting users at setting preferences. It proposes recommended preference values to the user as well as sets selected and learned preferences for services and application automatically.
- **User Configuration Manager** - This component realizes a central user configuration management for all user configuration information such as external service and application parameters, profiles, service credentials etc as well as CAMCAP user specific security, update, notification, submission, etc preferences.
- **Notifications, Triggers, Subscriptions** - This component allows the CAMCAP to offer a from the platform initiated data and information update service which sends out data and information notifications based on user specific and previously defined subscriptions and triggers. Examples for subscriptions could be users presence/context data, news feeds, special offers, buddy profile updates, Blog updates, RSS Feeds, community updates, etc. Examples for triggers could be business schedule triggers based on meetings, location triggers based on user locations etc.
- **Client Interface Layer** - This layer realizes the interface for all service and application clients. It builds an abstraction of the details of raw and higher level context data structures and provides access to them as well as to the Preference Learner, the User Configuration Manager and the Notifications component. In order to provide access to these components, the Client Interface Layer implements different access technologies and protocols such as SIP, Web Services (WS), other RFC based protocols (RFCs), a highly bandwidth optimized proprietary protocol (Prop.) and others.

V. IMS BASED PROTOTYPE

In order to proof, evaluate and demonstrate some of the concepts and features of the CAMCAP a first prototype with a limited feature set was implemented and a mobile social community application was built on top of it.

Due to the fact that IMS constitutes an All-IP infrastructure which allows the CAMCAP to interface any IP-based context data providing service and application on the Internet, IMS was chosen as development platform as well as underlying transport and signaling platform for providing the basic building blocks for connecting all community users, exchanging signaling and data messages and dealing with authentication and user registration. The Georgia Tech IMS lab was used as IMS platform, testbed and implementation platform and the CAMCAP as well as the social community application were implemented, deployed and tested there.

A. IMS Platform Architecture

Figure 2 shows an overview of the main parts (with reduced complexity) of the IMS specification compliant [10] system at the Georgia Tech IMS research lab. All components, protocols and interfaces shown in this figure are actually used by the CAMCAP and/or social community application prototype.

Due to the page limitations of this paper, component details could not be included at this point, please see [1] for a detailed description of the components, their functionalities and interworking.

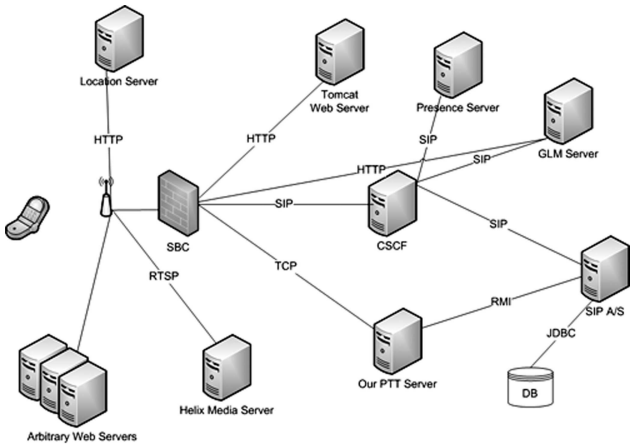

Fig. 3. IMS Social Network Application Architecture [1]

B. CAMCAP Implementation

The CAMCAP prototype was implemented on a SIP Application Server leveraging the SIP Servlet specification [11] and deployed and tested on the Georgia Tech IMS platform. The prototype does not aim to be complete or fully functional. Only a limited number of the proposed features and components and only the, for the social community application prototype implementation, essential ones were implemented. The following overview describes what features were implemented with what level of detail.

- **Data Aggregation Layer** - The Data Aggregation Layer supports SIP as data aggregation protocol and is able to process SIP Register, SIP Invite, SIP Publish and SIP Notify messages for IMS registration, session setup and presence update as well as SIP Message messages for a proprietary XML payload based location service solution provided by the Georgia Tech IMS lab.
- **Context Data Manager** - The Context Data Manager is realized by a MySQL 4.1 database with a simple OR-Mapper that transforms database data to JavaBeans and vice versa. The instantiated JavaBeans represent the raw context data managed in the prototype implementation. This raw data consists of Network Presence information about a user IMS registration(s), Service Presence information about a users IMS sessions, location information including location history, presence information, contact list data and community sets (only two levels supported).
- **User Configuration Manager** - The implemented User Configuration Manager allows CAMCAP users to specify presence status details, manage contact lists and define security and privacy parameters for context data such as location information, presence, etc.
- **Notifications, Triggers, Subscriptions** - The CAMCAP implementation allows users to subscribe to presence and location information updates of their contacts and the CAMCAP generates and sends out update notifications based on these subscriptions.
- **Client Interface Layer** - The Client Interface Layer implements a SIP interface that sends and receives SIP Message messages with CAMCAP specific XML payloads. For this implementation of the CAMCAP 21 different CAMCAP specific XML payload documents were defined. These XML documents enable clients and applications access to all CAMCAP data and features.

C. Social Community Application Implementation

This section gives and overview of the in [1] presented and on the CAMCAP based social community application that was implemented at the Georgia Tech IMS lab. It further discusses the application's dependencies on and interfaces to the CAMCAP. The implementation of the application was based on Nokia N80 mobile phones with J2ME, MIDP2.0, MMAPI and an IMS client SDK. It focused on a University campus community, so the CAMCAP Community Sets (see III-B) and application features like "Lectures" are specifically defined and designed for this setting. All other implemented features enable the application to be a full social community application with extended communication, collaboration and information features.

- **Map, Buddies and Hotspots** - One of the major goals at designing and developing this social community application was excellent usability and easy handling even though user input capabilities are very limited on mobile phones. The main user interface is a map of the geographically bounded community (see III-A). The left screen in Figure 4 shows the application's map interface. The map shows roads and landmarks of the community and visualizes two types of objects: Buddies and Hotspots. A user can browse through buddies and hotspots on the screen and select them in order to interact with them. The right screen in Figure 4 shows a buddy profile after a user selected the buddy on the map.
  Buddies are other community members a user added to one his/her own contact list. All buddies are rendered on the map and the location at which a buddy icon is rendered and what type of icon is used is tightly tied to location and presence. (See next item for more information)
  Hotspots are points of interests within the community. Basically there are no limitations to what a hotspot can be and in the implemented social campus community application types of hotspots range from restaurants, over landmarks and student centers to lecture halls and research labs.
- **Extended Presence and Location Information** - Location and presence information management is essential for every social community application
  The Georgia Tech location service offers an HTTP interface to determine the request sending device's location (GPS coordinates) based on its IP address and WLAN access point assignment. Since this is a first party location determination mechanism the social community application client needs to request the location information and use the CAMCAP SIP Message interface to publish the

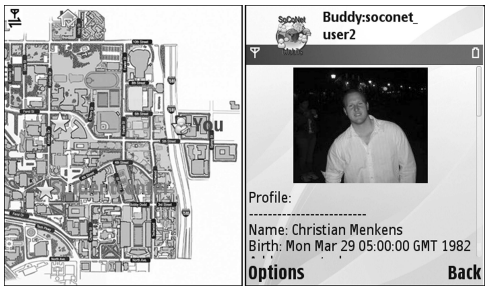

Fig. 4. IMS Social Community Application Phone Screens [1]

location information to the CAMCAP. The CAMCAP checks subscriptions, privacy and security parameters and generates and sends notifications to all clients that subscribed for location updates of this client.

For presence information, eight different presence/mood states (visualized with different icons on the map) where defined and are updated on the CAMCAP using the standard IMS SIP Publish messages. The CAMCAP checks subscriptions, privacy and security parameters and uses the standard IMS presence component to update subscribed clients using SIP Notify messages.

- **Contact Lists** - The application enables a user to create any number of contact lists on the standard IMS contact list server, to add contacts to these lists as well as to change the active/used contact list at runtime. All tasks performed at the contact list server are forwarded to the CAMCAP which stores references to and additional parameters for the contact lists.
- **Profiles** - A Profile contains data about a user or hotspot and is stored in the social community application database. This data transfer and rendering is done on application layer and no interface to the CAMCAP is necessary.
- **Communication** - Since a mobile phone is still mainly a communication device the social community application implements a representative set of communication and interaction features. All of these features are implemented on application layer and use the up-to-date presence and location information managed and provided by the CAMCAP. The features implemented are Push-To-Talk, Text Messaging / Chat, Picture Messaging, Voice / Video Call, Wall and Blog. For implementation details and IMS signaling flows see [1].
- **Multimedia** - Every user, hotspot and lecture has a media album that can contain links to video clips, live videos, pictures and audio clips and the social community application implements a player to view all different media types.
- **Lectures** - This feature shows the use of CAMCAP based Community Sets (see III-B). A user registers all lectures he/she takes and so, forms or joins community subsets that are part of the whole campus community set. By enabling to switch and limit the community view (contact list) to only the subsets, a user can communicate, collaborate and interact with the active users within the subset more efficiently and task oriented.
- **Events, Ticketing and News** - Hotspots can publish news and announcements and may offer and promote events/shows and allow users to purchase tickets for these events. Later extensions of the CAMCAP could include payment features and support this application feature on the application platform level.
- **Subscriptions and Notifications** - Users can subscribe to hotspot news, event information, announcements as well as blogs and walls of their contacts. These subscriptions and all corresponding notifications are managed on the CAMCAP.

VI. Conclusion and Future Work

This research work presented the design and architecture of a Context Aware Mobile Community Application Platform (CAMCAP) that consolidates and federates context information for all community applications built on top of it. The work presented a prototypical CAMCAP implementation based on an IMS development and deployment platform as well as an outline of a CAMCAP based prototypical implementation of an University campus social community application. This application showed that the CAMCAP supports developers at many community feature implementation tasks.

Future work will include user and developer studies that allow evaluating the development time advantage when using CAMCAP, the stability advantage as well as performance, security and usability factors. In future projects, the CAMCAP will be extended with more complex community features and more context data providers and client interface specifications will be implemented.

References

[1] C. Menkens, N. Kjellin, A. Davoust. IMS Social Network Application with J2ME compatible Push-To-Talk Service, International Conference and Exhibition on NEXT GENERATION MOBILE APPLICATIONS, SERVICES and TECHNOLOGIES (NGMAST), 2007.

[2] Anind K. Dey. Understanding and Using Context, Personal Ubiquitous Computing, 2001.

[3] J. Coutaz, J. L. Crowley, S. Dobson, and D. Garlan. CONTEXT is KEY, Communications of the ACM, 2005.

[4] The Facebook Project, http://www.thefacebookproject.com/wiki/index.php?title=Quick_Facts, 2008.

[5] University of Southern California - Information Sciences Institute, RFC791 - Internet Protocol, http://www.faqs.org/rfcs/rfc791.html, September 1981.

[6] J. Rosenberg et. al., RFC3261 - SIP: Session Initiation Protocol, http://www.ietf.org/rfc/rfc3261.txt, June 2002.

[7] B. Campbell et. al., RFC3428 - Session Initiation Protocol (SIP) Extension for Instant Messaging, http://www.ietf.org/rfc/rfc3428.txt, December 2002.

[8] 3GPP Organizational Partners, 3rd Generation Partnership Project; Location Services (LCS) (Release 7), 3GPP TS 22.071 V7.4.0, 2005.

[9] J. Rosenberg, RFC3856 - A Presence Event Package for the Session Initiation Protocol (SIP), http://www.ietf.org/rfc/rfc3856.txt, August 2004.

[10] 3GPP Organizational Partners, 3rd Generation Partnership Project; IP Multimedia Subsystem (IMS) (Release 7), http://www.3gpp.org/ftp/Specs/html-info/23228.htm, 2006.

[11] Sun Microsystems, Java Community Process, JSR116:SIP Servlet API, http://jcp.org/en/jsr/detail?id=116, 2003.