Making the World Wide Space Happen: New Challenges for the Nexus Context Platform

Ralph Lange, Nazario Cipriani, Lars Geiger, Matthias Großmann, Harald Weinschrott,
Andreas Brodt, Matthias Wieland, Stamatia Rizou, Kurt Rothermel
Universität Stuttgart, Universitätsstraße 38, 70569 Stuttgart, Germany
<firstname.lastname>@informatik.uni-stuttgart.de

***Abstract*—Context-aware applications rely on models of the physical world. Within the Nexus project, we envision a World Wide Space which provides the conceptual and technological framework for integrating and sharing such context models in an open, global platform of context providers. In our ongoing research we tackle important challenges in such a platform including distributed processing of streamed context data, situation recognition by distributed reasoning, efficient management of context data histories, and quality of context information. In this paper we discuss our approach to cope with these challenges and present an extended Nexus architecture.**

I. Introduction

Context-awareness is a major characteristic of pervasive computing. Context-aware applications rely on models of the physical world given by static context information such as map data and 3D models as well as dynamic information from billions of sensors located in our physical environment.

For both economic and technical reasons, it is highly desirable for such *context models* to be shared by a wide variety of applications. We envision a World Wide Space – an open, global platform in analogy to the WWW – which allows for integrating and sharing context models of commercial and non-commercial providers. Within the Nexus project [1], we research the conceptual and technological framework for such a system. The Nexus platform federates the context models of the different providers and offers context-aware applications a global, consistent view on their context data.

To show the vantages and necessity of federated context models we consider a use case with Cecilia, a PhD-student, traveling and preparing for PerCom in Galveston:

After the aircraft has reached normal flight altitude, Cecilia switches on her laptop and launches the *Context-Explorer*. It immediately discovers a context model provided by the aircraft's on-board entertainment system. The Context-Explorer visualizes the aircraft's interior stored in the model as well as the scenery below the clouds and the trajectories of the aircraft and nearby aircrafts recorded by radar.

Cecilia connects to the Internet and queries for context models containing her hotel in Galveston. The most detailed model is provided by the hotel company itself and secured by Cecilia's booking code. The Context-Explorer visualizes the 3D model enriched with dynamic information like the current positions of the elevators or the menu card in the lobby which promotes the daily special in the restaurant. Cecilia sends a context-aware message to all PhD-students entering the lobby within the next four hours and asks whether anybody of them wants to go out for dinner together.

Immediately after landing and entering the terminal building, Cecilia's PDA informs her about a WLAN-based positioning system and a context model for navigating to the exit. Cecilia launches the navigation application and requests navigation to her hotel using public transportation. The navigation application automatically connects to the required context models provided by the airport, the surveying office, and the bus operator. Cecilia does not need to hurry – the next bus will be nine minutes delayed as derived by reasoning on the latest movement of the bus and the current traffic on the highway, acquired by on-board GPS receivers and roadside units.

Just before arriving at the hotel, Cecilia receives an answer to her context-aware message: Tajel already spent three days in Galveston and knows a good pizzeria. In the evening, while dining, Tajel shares her personal context model acquired by her mobile phone with Cecilia. It provides Tajel's trajectory during sight-seeing, her geotagged photos, web-links acquired by RFID-readings at places of interest she visited, and the temperature profiles of the last days.

In the following, we present the Nexus platform for sharing and federating spatial context models to provide the technological fundamentals for such usage of context information at a global scale. In particular, we utilize the above use case for deriving novel challenges we tackle in our ongoing research. As a first result, we propose an extended architecture for the Nexus platform and present its components and approaches.

The remainder of the paper is structured as follows: In Section II we present the basic approach in Nexus and our previous works, before we derive new challenges in Section III. In Section IV we propose an extended architecture for Nexus and describe the different approaches to tackle those challenges. Then, we discuss related work in Section V. Finally, the paper is concluded in Section VI.

II. Nexus Approach

To implement the vision of a World Wide Space, centralized context management systems are obviously insufficient; a scalable, distributed architecture is required. In this section, we briefly describe the architecture and the data model of the current Nexus platform [1] which serves as the starting point for the extensions required to handle the above scenario.

Published in Proceedings of the 7th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '09), pp. 300-303. Galveston, TX, USA. March 2009.

http://dx.doi.org/10.1109/PERCOM.2009.4912782

Nexus uses a three layer architecture [2], where applications are located on the top layer. The middle layer forms a federation, which integrates the data stored on context providers on the bottom layer. Nexus follows a request-response model receiving queries from applications. Based on spatial restrictions in the query, the federation layer determines relevant context providers and forwards the query to them. In a second step, it integrates the results and sends them back to the application.

Depending on the type of data to be hosted, different implementations for context providers are necessary [3]: For static, complex data, like building outlines, off-the-shelf RDBMSs with spatial extensions and Nexus-specific wrappers are a reasonable option. For volatile data, like sensor measurements, main memory databases are a better choice [4]. Position data of mobile objects requires context providers supporting a hand-over mechanism for objects to be able to handle objects crossing the boundaries of service areas [5].

Nexus provides an extensible data model based on object-oriented concepts, the Augmented World Model (AWM) [6]. The AWM consists of the Standard Class Schema (SCS) and Extended Class Schemes (ECS). The SCS defines the root type of the type hierarchy and a set of types we consider relevant for most context-aware applications, such as buildings, rooms, and train stations. Context providers requiring more specialized types can define ECSs containing sub-types of SCS types. As every object of an ECS type can be cast to an object of a SCS type, those objects are at least partially useful for applications not knowing the ECS.

III. New Challenges

The above traveling scenario highlights several novel requirements to scalable context management – not only beyond the capabilities of the current Nexus platform but also new research challenges in context-aware computing.

Scalable stream-processing of heterogeneous context. Visualizing the aircrafts' trajectories and the scenery below the clouds is an example for continuous processing of highly dynamic together with static context data from different sources.

A scalable context management platform such as Nexus requires a comprehensive, distributed processing framework considering the diverse characteristics of context data. First, a novel processing model integrating static context data and data streams which extends the state-of-the-art data models for stream processing (e.g., [7], [8]) has to be designed. Subsequently, distribution mechanisms for scalable data processing within the platform have to be researched. Moreover, the processing framework has to be open for new classes of context data and domain-specific operations.

Distributed situation recognition. The delay of Cecilia's bus is an example of a complex situation within the real world which cannot be acquired directly from few sensors but which has to be derived from expert knowledge and many sensor readings from the bus's GPS receiver and roadside units.

While existing approaches for situation recognition base on one or several independent centralized systems (e.g., [9], [10]), global context management requires novel distributed reasoning algorithms as the input data generally is stored in different context models. Moreover, heuristics for distributing a given reasoning task to various providers optimizing communication cost and timeliness of situation information are required. For re-use of recognized situations such as a traffic jam, concepts for hierarchical situation recognition have to be researched.

Temporal aspects and data histories. Cecilia's personal context model acquired by her mobile phone provides a lot of information on the past such as her trajectory during sightseeing and RFID-readings.

Managing histories of diverse context data poses several new challenges to the Nexus platform and context management in general: Predicates and query operators for time-based access have to be defined for federated context models. The schema has to be extended for temporal aspects – not only regarding attribute values but also regarding relations between objects. To save storage capacity, data reduction algorithms are required. In particular with moving objects these algorithms also have to minimize cost for wireless communication and allow for tracking in real-time.

Managing moving objects' trajectories in an global context management platform requires distributed spatiotemporal index structures, beyond existing ones for centralized database systems [11], since the objects move between the providers' service regions.

Context-aware workflows. During her trip Cecilia uses many different Nexus services, such as querying context data about the hotel and the plane, or requesting navigation support at the airport. For future journeys, Cecilia should be able to design a travel workflow describing her standard order of service usage, instead of selecting the services manually over and over again.

Thus, the Nexus platform has to allow for predefining desired flows using Nexus services or other web services to enable a service-oriented development of context-aware applications.

Quality of context information. Context data generally is subject to inaccuracies since it is mostly acquired by sensors.

The openness of the Nexus platform requires powerful concepts for specifying the accuracy of different classes of context data in a generic but consistent manner which is compatible with existing, specialized approaches for individual classes of context data such as GPS position data. Also, algorithms for fusion of inaccurate context data from different context providers on the same phenomenon have to be researched. At the same time, these algorithms must factor in dependencies between sensor readings. For instance, the absolute accuracy of the trajectory data on nearby aircrafts in the above scenario not only depends on the on-board radar of Cecilia's aircraft but also on the accuracy of the position data on this aircraft – possibly acquired by GPS.

Therefore, the Nexus platform requires a reference model for integrating context data from different sources considering accuracy of context data, inconsistencies, and reliability of context providers.

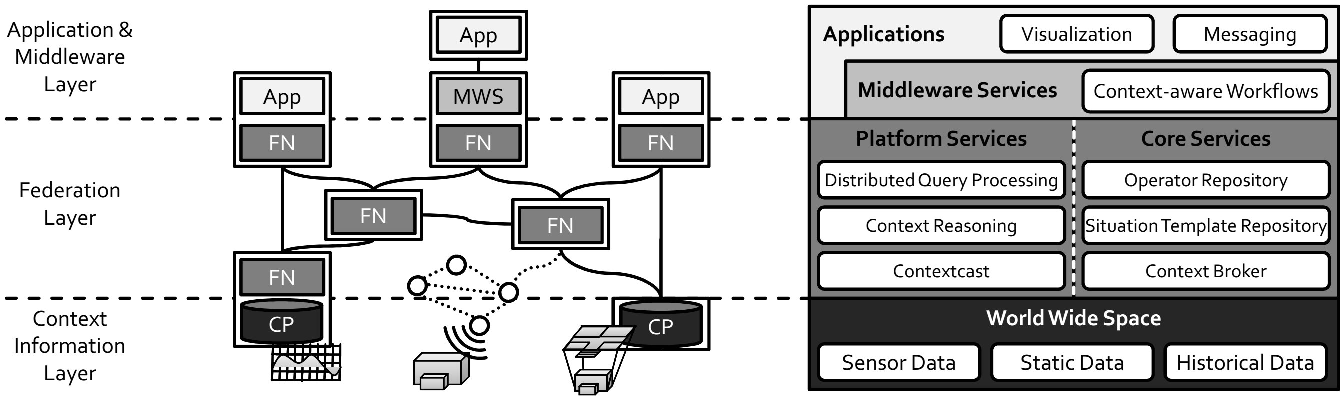

Fig. 1. Extended Nexus Architecture

IV. Extended Nexus Architecture

The extended Nexus platform retains the idea of separating applications, federation, and providers of context data. However, the extensions add more flexibility to the federation layer and integrate historical data within the context data layer. Figure 1 depicts the extended Nexus architecture with its three layers.

Next, we discuss the layers from bottom to top as well as the Nexus reference model for quality of context information.

A. Context Information Layer

The *Context Information Layer* consists of context data servers from arbitrary providers (CP). The CPs provide context data at different levels of detail ranging from sensor data over static context to data histories and constitute the World Wide Space.

Historical data, such as trajectories of moving objects or value patterns of temperature sensors, are stored by specialized history servers. In [12], [13], and [14] we propose online data compression algorithms for trajectories to reduce the amount of data to be stored. The context data from history servers can be exported to history warehouses for more sophisticated analysis such as traffic jam prognosis.

B. Federation Layer

The *Federation Layer* is a distributed platform for context services provided by federation nodes (FN). The FNs can be servers of infrastructure-based networks as well as mobile devices [15] and compose a hybrid network, cf. Figure 1.

There are two basic classes of Nexus services: *Platform Services* are context services typically used by applications, such as *Context Reasoning*, *Context Cast*, and *Distributed Query Processing*. In contrast, *Core Services* provide the functionality on which the Nexus Platform Services rely.

Distributed Query Processing. Streamed data is highly volatile, potentially infinite, and allows only sequential access. This requires dedicated stream processing functionality to enable on-the-fly processing of streamed data. Furthermore, to address the challenges described in Section III, it is essential to combine streamed and static data. The **Operator Repository** enables stream processing, providing suitable data stream operators. To avoid load congestion on a particular site, partitioning and distributing queries across processing nodes is an essential step to make stream processing scalable.

Context Reasoning. Context reasoning derives new knowledge from low-level context. Nexus uses a situation-centric approach describing each situation by a set of rules which constitutes a *Situation Template* stored in the **Situation Template Repository**. Each Situation Template generates a logical execution plan, a directed graph describing the data flow and the steps of the algorithm. Currently, we use Classic Logic and Bayesian Networks as reasoning algorithms.

For scalability reasons, the situation recognition process must be distributed over several physical nodes. The distribution of the logical execution plans to physical machines are governed by factors such as latency, bandwidth, and load.

Contextcast. The Contextcast service enables applications and services to send messages to entities with a certain context. Message distribution does not rely on explicit multicast groups, but uses an overlay network of context-based routers to forward messages instead. This approach is similar to content-based publish/subscribe systems, however, the forwarding structures are adapted to exploit properties of context information such as gradual changes and support for location information.

Context Broker. The Context Broker discovers relevant CPs for query processing and context reasoning. It indexes all CPs by means of their models and allows for querying for relevant providers whose models intersect a certain clipping of the federated context model.

Moreover, it provides distributed index structures [16] for accessing trajectory data on moving objects since these objects are not bound to a specific CP and their trajectory data may be distributed over many providers.

C. Applications & Middleware Layer

The *Applications & Middleware Layer* consists of context-aware applications as well as middleware services (MWS). For example, parts of the application logic can be outsourced into the MWS as **Context-aware Workflows** enabling the users to orchestrate repeating service usage [17], [18].

D. Reference Model for Quality of Context Information

The reference model for quality of context information provides a conceptual framework for describing the accuracy of diverse classes of context data. It shall also allow for formalizing inconsistencies between context models and for specifying the reliability of CPs.

In [19] we propose a powerful approach for extending context information with meta data – particularly on data quality. In [20] and [21] we present energy-efficient protocols for monitoring moving objects with well-defined accuracies for different types of spatiotemporal queries.

V. Related Work

Context-aware systems have been researched thoroughly and are appearing on the consumer market. However, existing services, such as intelligent navigation systems or health observation systems, are limited to one particular scope. In contrast, Nexus is designed to provide a truly global, highly dynamic, federated model of the world. The sheer amount of data for such a model requires an open approach, with many collaborating context providers.

Aura [22] and SenseWeb [23] are projects with similar goals, focusing on selected classes of context information, namely technical context and sensor data, respectively.

Over the past decade many research efforts have been made in the field of data-stream processing. Prominent examples are Borealis [7] and STREAM [8]. These systems provide rich, general purpose stream-processing functionality, but do not exploit context-specific properties for query optimization or execution speed-up. Most spatio-temporal indexes are designed for centralized systems only [11]; an approach for processing range queries in distributed spatio-temporal databases was presented in [24], which complements our work [16]. Existing reasoning approaches are either based on centralized systems [9] or on independent sub-systems [10]. In Nexus, we aim at a scalable reasoning system, exploiting data locality as well as intermediate results. Workflows [25] have also been a research topic over the last years. Nexus is working on an extension for BPEL [26] to flexibly incorporate context data from the World Wide Space into the description of workflows.

To the best of our knowledge, Nexus is the first context-aware system to incorporate these techniques.

VI. Conclusions

In this paper we presented open challenges for context-aware platforms beyond the distributed management of open and global world models. We explained amongst others the impact of distributed processing of streamed context data, the efficient management of context data histories, and situation recognition by distributed reasoning.

We presented the extended architecture of our Nexus system to tackle these new challenges. Moreover, we presented first approaches and results towards the World Wide Space.

Acknowledgements

This work is partially funded by the German Research Foundation within the Collaborative Research Center 627.

References

[1] Collaborative Research Center (SFB) 627. [Online]. Available: www.nexus.uni-stuttgart.de

[2] D. Nicklas, M. Großmann, T. Schwarz, S. Volz, and B. Mitschang, "A model-based, open architecture for mobile, spatially aware applications," in *Proc. of SSTD*, 2001.

[3] M. Großmann, M. Bauer, N. Hönle, U.-P. Käppeler, D. Nicklas, and T. Schwarz, "Efficiently managing context information for large-scale scenarios," in *Proc. of PerCom*, 2005.

[4] M. Bauer, C. Becker, J. Hähner, and G. Schiele, "ContextCube—providing context information ubiquitously," in *ICDCS Workshops*, 2003.

[5] A. Leonhardi and K. Rothermel, "Architecture of a large-scale location service," in *Proc. of ICDCS*, 2002.

[6] D. Nicklas and B. Mitschang, "On building location aware applications using an open platform based on the NEXUS augmented world model," *Software and System Modeling*, vol. 3, no. 4, pp. 303–313, 2004.

[7] D. J. Abadi, Y. Ahmad, M. Balazinska, U. Cetintemel, M. Cherniack, J.-H. Hwang, W. Lindner, A. S. Maskey, A. Rasin, E. Ryvkina, N. Tatbul, Y. Xing, and S. Zdonik, "The Design of the Borealis Stream Processing Engine," in *Proc. of CIDR*, 2005.

[8] A. Arasu, B. Babcock, S. Babu, M. Datar, K. Ito, R. Motwani, I. Nishizawa, U. Srivastava, D. Thomas, R. Varma, and J. Widom, "Stream: The stanford stream data manager," *IEEE Data Eng. Bull.*, vol. 26, no. 1, pp. 19–26, 2003.

[9] H. Chen, T. Finin, and A. Joshi, "Semantic web in the context broker architecture," in *Proc. of PerCom*, 2004.

[10] A. Ranganathan and R. H. Campbell, "A Middleware for Context-Aware Agents in Ubiquitous Computing Environments," in *Proc. of Middleware*, 2003.

[11] M. F. Mokbel, T. M. Ghanem, and W. G. Aref, "Spatio-temporal access methods," *IEEE Data Eng. Bull.*, vol. 26, no. 2, pp. 40–49, 2003.

[12] N. Hönle, M. Großmann, D. Nicklas, and B. Mitschang, "Preprocessing position data of mobile objects," in *Proc. of MDM*, 2008.

[13] R. Lange, F. Dürr, and K. Rothermel, "Online trajectory data reduction using connection-preserving dead reckoning," in *Proc. of MobiQuitous*, 2008.

[14] R. Lange, T. Farrell, F. Dürr, and K. Rothermel, "Remote real-time trajectory simplification," in *Proc. of PerCom*, 2009.

[15] D. Dudkowski, H. Weinschrott, and P. J. Marrón, "Design and implementation of a reference model for context management in mobile ad-hoc networks," in *Proc. of AINA Workshops*, 2008.

[16] R. Lange, F. Dürr, and K. Rothermel, "Scalable processing of trajectory-based queries in space-partitioned moving objects databases," in *Proc. of ACM GIS*, 2008.

[17] M. Wieland, O. Kopp, D. Nicklas, and F. Leymann, "Towards Context-Aware Workflows," in *Proc. of CAiSE*, 2007.

[18] M. Wieland, D. Nicklas, and F. Leymann, "Managing technical processes using smart workflows," *ServiceWave*, Dec. 2008, to appear.

[19] N. Hönle, U.-P. Käppeler, D. Nicklas, T. Schwarz, and M. Großmann, "Benefits of integrating meta data into a context model," in *Proc. of PerCom Workshops*, 2005.

[20] T. Farrell, R. Lange, and K. Rothermel, "Energy-efficient tracking of mobile objects with early distance-based reporting," in *Proc. of MobiQuitous*, 2007.

[21] T. Farrell, R. Cheng, and K. Rothermel, "Energy-efficient monitoring of mobile objects with uncertainty-aware tolerances," in *Proc. of IDEAS*, 2007.

[22] G. Judd and P. Steenkiste, "Providing contextual information to pervasive computing applications," in *Proc. of PerCom*, 2003.

[23] L. Luo, A. Kansal, S. Nath, and F. Zhao, "Sharing and exploring sensor streams over geocentric interfaces," in *Proc. of ACM GIS*, 2008.

[24] G. Trajcevski, H. Cao, P. Scheuermann, O. Wolfson, and D. Vaccaro, "Online data reduction and the quality of history in moving objects databases," in *Proc. of MobiDE*, 2006.

[25] F. Leymann and D. Roller, *Production workflow: concepts and techniques*. Prentice Hall, 1999.

[26] *Web Services Business Process Execution Language Version 2.0 – OASIS Standard*, OASIS, 2007.