Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 1. pp. 283–293.

PECES – PErvasive Computing in Embedded Systems

Zoltán Rak, Vilmos Bilicki

Frontendart Ltd., Szeged, Hungary
e-mail: rakz@frontendart.com, bilickiv@frontendart.com

Abstract

The dramatic growth of the amount of information that is made available through computer systems and the increasing need to access relevant information anywhere at any time are more and more overwhelming the cognitive capacity of human users. Thus, instead of providing the right information at the right time, current computer systems are geared towards providing all information at any time. This requires humans to explicitly and repeatedly specify the context of the required information in great detail.

The overall problems resulting from this type of information access are amplified by the fact that an ever-increasing number of users are accessing information on-the-move through portable computer systems such as PDAs and cellular phones. These systems are becoming increasingly ill-suited to provide efficient mobile access to relevant information.

The vision of Pervasive Computing aims at solving these problems by providing seamless and distraction-free support for user tasks with devices that are invisibly embedded into the environment. In order to provide task support in an unobtrusive and intuitive way, the devices are equipped with wireless communication and sensing technology. This allows them to cooperate with each other autonomously, i.e. without manual intervention, and it enables them to perceive relevant parts of the physical world surrounding their human users.

Keywords: pervasive computing, m2m systems, domain independent architecture, middleware, context ontology, smart spaces, security, development tools

1. Beyond the state of the art

In the beginnings of the Pervasive computing software architectures such as GUIDE system were developed and deployed together with platforms that were designed exactly for the needs of the target applications. The importance of the context

information very recognized very early at the development of pervasive applications [15]. However traditional software architecture designs do not use the context information opposite to PECES design where it is utilized inside the middleware to improve the functional parts during cooperation between different entities in the system. This paper will outline how the ontologies as basis of the system's model description will actively contribute in the dynamic group forming, communication and the security issues as well. There are already middleware designs which aim at forming specific smart environments so called restricted smart areas or "islands of integration". They focus on specific types of smart environments such as meeting or class rooms [4, 11] or dynamically constructed environments on the basis of proximity [1, 6, 10, 14, 17]. Opposed to this PECES project creates a general layer with no specific application scenario and restrictions to exact location.

2. Technical objectives and innovation

The features outlined above will be detailed through real life use cases in the upcoming sections to prove the benefit that PECES adds to development of Machine2Machine [7] systems. To go a step further PECES also defines so called development tools to provide an environment for developers where they can design and test PECES based software environment even before a first real deployment to the target platforms. The real life uses case on which the research concepts will be mapped is a "Nursling Care" system where a telemedicine solution is introduced-developed by Frontendart Ltd. [3]-which takes care of patients and monitors them. This architecture will completely reside on PECES middleware. FEA has a role in this scenario as Machine2Machine technology provider outlined in the paragraph below.

From the technology and business model point of view, the introduced "Nursling Care" scenario is built on the M2M service provider infrastructure, which provides access to the last mile of services for different domain specific service providers. Machine2Machine (M2M) solutions provide a new way of monitoring or even controlling the workflows in different domains. The actual M2M solutions are mostly domain specific and they are not extendable or reusable in different application domains. This issue is the same mentioned at the introduction when formation of specific environments was outlined. In most cases, it is not feasible for the end users or the service providers to install and maintain dedicated solutions for each covered domain. A more feasible approach could be the introduction of the general purpose M2M infrastructure provider (there could be multiple providers) who is going to provide the basic M2M infrastructure. On the top of this infrastructure, domain specific devices and services could be deployed and offered to the end users as it is shown on the figure below. This paper proves that PECES totally fits to these needs.

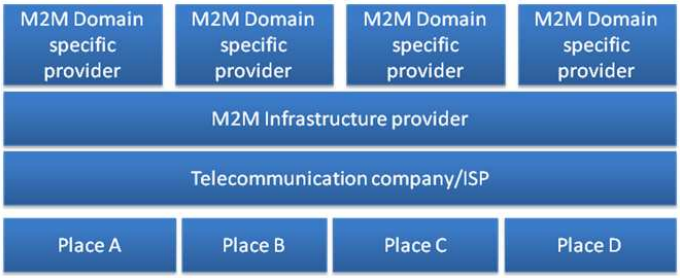

Figure 1: M2M Ecosystem

3. Domain model specification

3.1. Challenges at Nursling Care

As said M2M services are domain independent solutions able to hold multiple domains on the top of them. Every system specification's core part is the definition of the domain model. Using context ontologies enables effective context description and distribution of information in a generalized way. The benefit here opposite to common development solutions is that the developers have a common vocabulary to share context information which can be interpreted by the entities participating in a specific domain. In fact the domain model of a software product can be fully described with these definitions. This can be imagined as not only the attributes of different entities are specified with this description language but the relationship and the information flow between them. Later these definitions will also be used for determination of group members in the specific and dynamically formed Smart Spaces. In "Nursling Care" use case there will be intelligent vital sign measurement devices, intelligent household devices, PDAs taken by the visiting nurses which all have to be grouped in the right way. The definition of the context can be imagined as the definition of the logical system plan where the main task is to define the domain model of the system.

3.2. PECES solutions and practice

The tool used for ontology handling which will be integrated in the mentioned development tools is Protégé. This is a java based tool with a graphical user interface which enables easy handling of ontology definitions. Beside this an extra feature is the visualization of context relationships which is provided by the Ontoviz plugin [8]. As an example in the "Nursling Care" domain a part of the context of the blood pressure measurement device can be described with the following ontology definitions:

```
<participatingDevice>
        <p2:Member rdf:ID="myMeasuringSensor0">
              <hasContext rdf:resource="#installedToMySmartHome0"/>
```

```
            <p2:measures>
                  <p3:DiastolicBloodPressure rdf:ID="myBabyBloodPressure0"/>
            </p2:measures>
            <hasContext rdf:resource="#relatedToeHealthBabyCare0"/>
            <ownedBy>
                  <user:User rdf:ID="myParent0">
                        <owns rdf:resource="#myMeasuringSensor0"/>
                        <owns rdf:resource="#myHomeHub0"/>
                  </user:User>
            </ownedBy>
            ...
</participatingDevice>
```

The visualisation of this definition can be easily done with the Ontoviz plugin which is present on the figure below showing the relationships between entities.

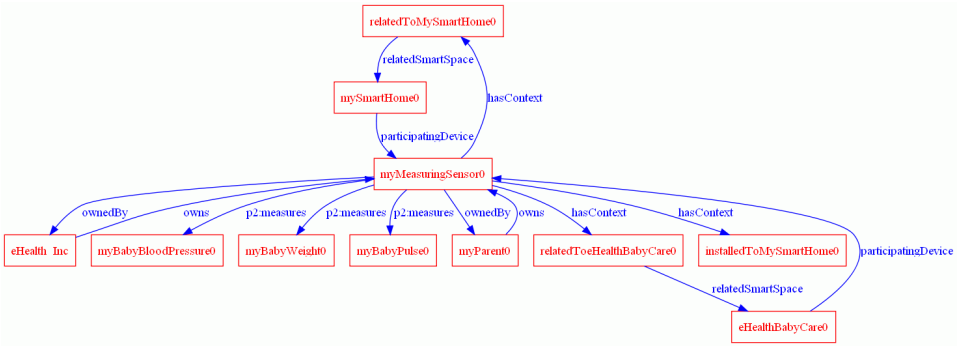

Figure 2: Measurement Sensor Description

There are predefined ontologies which define base and common concepts and are namely Smart Space, Device profile, User profile, Service and Event definitions. These are the so called core ontologies which are always the basis of the defined model. The developers can then add software specific definitions which concentrate multiple or extend one of these predefined entities.

In our example there are two services, the "eHealthBabyCareService" and "eHealthProviderService". The relations between entities mentioned above can be defined easily by using the formalism from the ontologies. It provides properties as it can be seen on the previous figure which names are "ownedBy", "measures" "hasContext" and further ones like "serviceProvidedBy", "serviceConsumes", "serviceConsumedBy". These properties allow for the developers to identify the relationships; devices at patient's home (like weight scale or blood pressure measurer) which consume the "eHealthBabyCareService" and a PDA carried by the nurse which consumes both services-depending on where it is located-to calculate the visit route to patients and later gather information at patient's home.

The advantage of using this formalism is not only that the developers have effective context description tools but with this they already define the domain model of their system and have effective and rapid development lifecycle. This

context information will be also actively used by the middleware during runtime to deal with the grouping, communication and security concepts detailed in the upcoming sections.

4. Communication, Dynamic Grouping, Smart Spaces

4.1. Challenges at Nursling Care

The M2M system consists of distributed entities which have different roles, from vital sign measurement to uplink provision to the central database. The entities must have and effective way of topology formation to have a clear global architecture of the system. The "Nursling Care" scenario have so called Home Hubs which are the main access points in the Smart Homes and provide uplink to the outer World, also there are the mentioned measurement devices and other private household property. All these entities have to be separated in logical groups where they belong. PECES has an exact mechanism for this purpose which uses the context ontologies to determine these groups. This section will start with introducing the usage of the ontologies during runtime for logical decisions.

4.2. PECES solutions

PECES middleware will be an extension of the already existing BASE middleware [1] mentioned before. It'll have modular architecture with inter-process communication, network communication, context provisioning, registry and role assignment mechanism and many more extensible parts. The modular architecture is useful because of the resource insufficiency problem at embedded devices. Devices with fewer resources will have a lightweight version of middleware and will be helped out by more powerful devices. As mentioned context ontologies will be also used as description for group formation. Two important parts have to be mentioned here namely the Contex Provisioning and Role Based Group Formation or Role Assignment Mechanism.

The grouping concept has its well defined actors. These are the Coordinator who holds the information about what context is needed to join a Smart Space (Logical Group), the Gateway which has extended communication capabilities namely an uplink to Internet and provides the connection to other Smart Spaces and last but not least the plain Members, devices which have the context needed to join the Smart Space. Another important thing is the registry where the contextual information is distributed. The registry is hierarchical and it has three types the Device level, Space level and Internet level registry. Devices which provide services export this information to different levels of registries and later consuming devices can find them. Gateways are separated into two further groups of Local and Remote Gateways. Remote Gateways are the classical gateways, Local Gateways have the purpose two seamlessly connect devices with incompatible interfaces, as they have

multiple different type interfaces. This information also needs to be described in the devices' context in order to be available for other devices. The following figure will help to understand the essence of the middleware's modular architecture and show how this mechanism works.

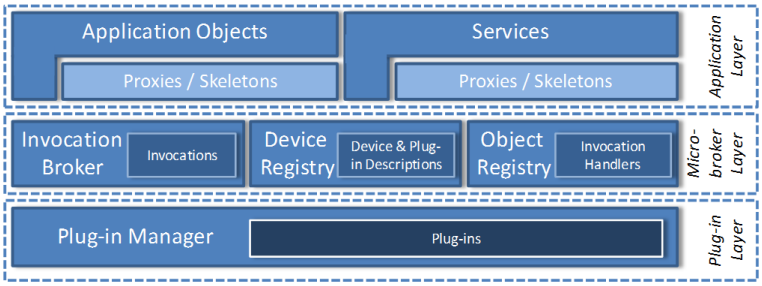

Figure 3: BASE Architecture

As it can be seen the core part is the micro broker layer. It have three main parts, the invocation broker the device registry and object registry. The invocation broker is responsible for forwarding and dispatching invocations. Simply said applications register for interesting events here and the broker indicates if the event occurred and provides the needed information. Registry holds the contextual information about the device and the plug-ins. The object registry stores the information about the registered objects handled by the invocation broker. Plug-ins are presenting the extensible manner of the architecture they can be serializer-deserializer plug-ins, security plug-ins (encryptor-decrpytor), transceiver plug-ins etc. All these plug-ins have the purpose-after they are installed and ordered in a proper sequence-to provide seamless integration in the physical world surrounding the device and provide automated communication interface with the other entities. Transceiver plug-ins can be for example network communication stacks for different types of protocols and interfaces.

4.3. PECES in practice

The "Nursling Care" use case will show how this mechanism will work in practice. As it was mentioned at the beginning of this section in this scenario there are multiple devices used for different purposes. The example Smart Space, actually two Smart Spaces-this exactly shows how multiple domains can be injected on the same infrastructure-will be the eHealthBabyCare and SmartHome which are formed at same home (physical topology) separating different domains of use. The Smart Space borders show on the figure on the next page, how the devices are grouped based on the context they have. The nurse who carries her business PDA and visits the patients enters the house, and after the discovery and exchanged contextual information joins to eHealthBabyCare Smart Space is. However there is a bigger SmartHome Smart Space which members are unreachable for the nurse's PDA.

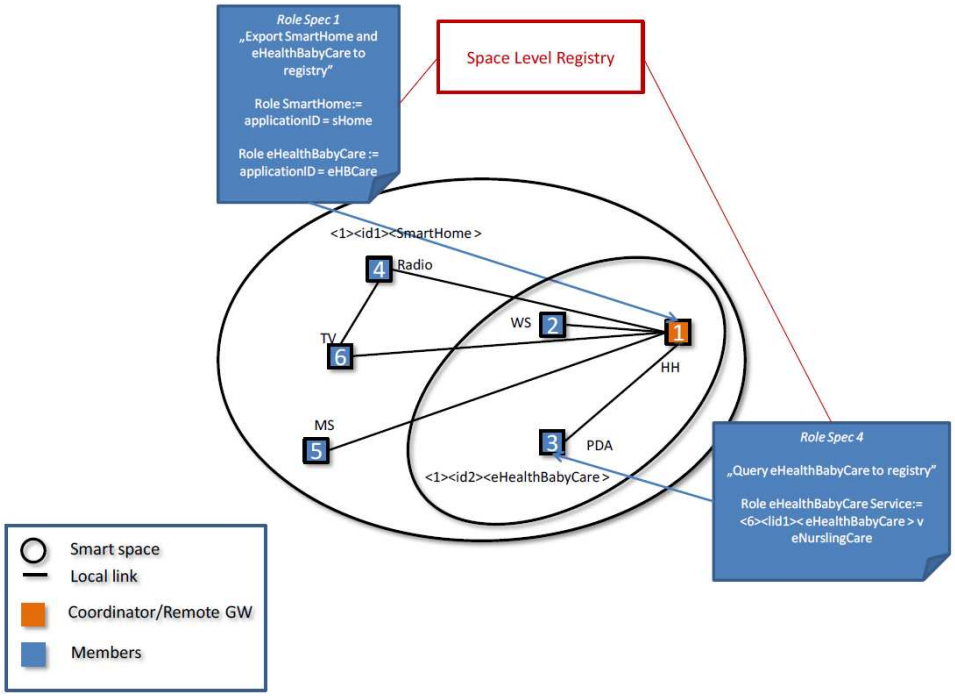

Figure 4: Dynamic Group Formation

As seen even hierarchical and multiple overlapping Smart Spaces can be formed by defining different contexts and the logical boundaries will be built seamlessly. The question here is what if the nurse's PDA says it has a context which grants join to SmartHome space. The answer for this question will be detailed in the upcoming section where the trust relationships and certificate hierarchies will be associated with grouping concepts. To stay at grouping mechanisms the simple sequence of the group formation is the following. Every PECES enabled device has its context which describes it. The Smart Spaces are defined by a set of rules which are in fact constraints on context; this role (or rule set) is stored on the Coordinator as mentioned before. Simple rule set is demonstrated in the listing below:

```
eHealthBabyCare == hasContext="eHealthBabyCare0" ^ relatedTo="EhealthBabyCare0"
^ (ownedBy="myParent" v ownedBy="eHealthIncompany")
```

As an engine the Coordinator has a role assignment mechanism which determines whether a device who wants to join the Smart Space has the needed context or not. The group formation and communication will be done seamlessly by the middleware layer. The applications and services residing on the top of the middleware will see an abstract high level picture about the world surrounding them, in form of available services and entities. The "Nursling Care" scenario here has eHealthBabyCareService which is exported to the Space Level registry on the Coordinator so the measurement devices can find it after they have joined the Smart Space

and can send vital sign measurement information about the monitored babies or patients which can be later queried by the nurse's PDA or sent to the central.

5. Security concepts

5.1. Challenges at Nursling Care

The base concept we are talking about is the M2M infrastructure based "Nursling Care" telemedicine solution which circulates a lot of sensitive and private information. Medicine and also personal information handling is always very critical and requires a big responsibility how to handle it. However nowadays security aspect is important at almost any kind of network based software design. There are two main things security needs to cover here. The first one is to protect our data from being captured by third party persons and the second to have standardized mechanism for building trust relationships. Actually the security concept will be injected to the middleware in a form of a plug-in as it was already mentioned at the previous section.

5.2. PECES solutions and practice

The basis of PECES security can be outlined with three main segments which are the trust, key management and authentication concepts. PECE has a very simple but effective trust relationship model. It has three main types: "non", "marginal" and "full". Based on this effects caused to the device's system are separated to "marginal" and "critical" effects. So a trust relationship table can be defined where peers with full trust are granted to do every kind of interaction, peers with marginal trust are granted to make only things that cause marginal effects and non trusted parties are fully locked out from the corresponding device. As mentioned before role assignment mechanisms can be validated with this concept if we use certificates as the credentials for trust relationship.

PECES middleware will rely on the asymmetric cryptography [5] during the basic key setups. After the symmetric session keys were exchanged faster symmetric encryption will be used during the communication. This combination of two cryptographies provides a safe and fast system at the same time. As it was said before circulation of certificates will handle the trust relationships, public key part of the asymmetric key pair can be also signed with another key pair with this creating a recursive certificate hierarchy. Simply explained the figure below device that has the SmartHomeCert certificate is granted full access to the Smart Home but the visiting Nurse's PDA will only have the eHealthBabyCareCert certificate which will grant access only restricted areas where the "baby care" related entities are residing. In other words she won't be able to access patient's private property like tuning the intelligent TV or setting the alarm system.

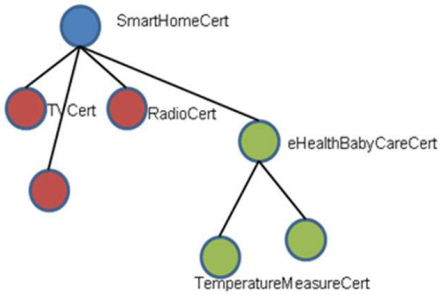

Figure 5: Certificate Hierarchy and Trust

PECES middleware has its dedicated key storage for storing the own asymmetric key pair, the certificates from other parties and the symmetric session keys generated during communication establishment. The security plug-in is fed from this dedicated key storage. Therefore the authentication concepts consist of two parts the device authentication which is done by the proper certificates and the data authentication which is provided by the initial asymmetric cryptography and after the session keys are exchanged with symmetric cryptography. These carefully designed security concepts provide high level authenticity and integrity and are excellent choice for protecting the sensitive patient and medicine related private data.

6. Development process

Last but not least a challenge not just here but at every development process is the way and tools used to develop the imagined system. PECES will offer an Eclipse [2] based development environment extended with its own plug-ins called Development Tools. There will be three tools namely the Configuration, Modelling and Testing tool. With the Configuration tool the developer can handle and define the initial context information, this will provide a Protege like GUI where ontologies can be easily defined and maintained. Modelling tool will help in definition of the system's model (Smart Spaces, Roles, Middleware components), and as the last one Testing tool will provide a test environment which is simulating real circumstances and where developers can test the developed system even before they deployed their software to real platforms. These tools are all OSGi [9] bundles and will integrate seamlessly to Eclipse IDE, providing a solution for easy, effective and fast software development for the developers.

Conclusions

This paper was about to outline what is the benefit of adapting PECES on software development of wide range of software products and how it fits to the vision of domain independent M2M architectures. The aim was to identify its general manner

and to introduce the power of bringing the outlined technologies together. There is no question about PECES will deliver a next generation, new way of integrated software development and gives the ability to developers to do what they know the best and this is the application and service development itself, having no troubles with infrastructure and integration issues.

References

[1] C. Becker, G. Schiele, H. Gubbels, K. Rothermel: BASE - A Micro-broker-based Middleware For Pervasive Computing. 1st IEEE International Conference on Pervasive Computing and Communications (PerCom 03), pp. 443-451, March 23-26, Fort Worth, USA, 2003.

[2] http://www.eclipse.org/

[3] http://www.frontendart.com

[4] M. Román, C. K. Hess, R. Cerqueira, A. Ranganathan, R. H. Campbell, K. Nahrstedt: Gaia: A Middleware Infrastructure to Enable Active Spaces. IEEE Pervasive Computing, pp. 74-83, Oct-Dec 2002.

[5] CCITT, Recommendation X.509, "The Directory-Authentication Framework", Geneva, 1989

[6] E. Aitenbichler, J. Kangasharju, M. Mühlhäuser: Experiences with MundoCore. 3rd IEEE Conference on Pervasive Computing and Communications (PerCom'05) Workshops, Kauai Island, Hawaii, USA, pp. 168-172, 2005.

[7] http://en.wikipedia.org/wiki/Machine_to_Machine

[8] http://protegewiki.stanford.edu/wiki/OntoViz

[9] http://www.osgi.org/Main/HomePage

[10] C. Becker, M. Handte, G. Schiele, K. Rothermel: PCOM - A Component System for Adaptive Pervasive Computing Applications. 2nd IEEE International Conference on Pervasive Computing and Communications (PerCom), 2004, Orlando, USA.

[11] U. Saif, H. Pham, J. M. Paluska, J. Waterman, C. Terman, S. Ward: A Case for Goal-oriented Programming Semantics, Workshop on System Support for Ubiquitous Computing at the 5th Annual Conference on Ubiquitous Computing (UbiComp), 2003.

[12] http://www.ict-peces.eu/

[13] http://protege.stanford.edu/

[14] A. Ferscha, M. Hechinger, R. Mayrhofer, R.Oberhauser: A Light-Weight Component Model for Peer-to-Peer Applications. 2nd International Workshop on Mobile Distributed Computing, March 2004.

[15] B. Schilit, N. Adams, R. Want: Context-Aware Computing Applications. Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, pp. 85-90, December 1994.

[16] http://en.wikipedia.org/wiki/Telemedicine

[17] N. Kawaguchi: VPcogma: A Light-Weight Cooperative Middleware for Ubiquitous Embedded Devices, Workshop on Software Architectures for Self-Organization at 3rd International Conference on Pervasive Computing (Pervasive), 2005.

Zoltán Rak, Vilmos Bilicki
Hungary, Szeged, Frontendart Ltd., Zászló utca 3.