Design of a Distributed System using Mobile Devices and Workflow Management for Measurement and Control of a Smart Home and Health

Pawel Czarnul
Faculty of Electronics, Telecommunications and Informatics
Gdansk University of Technology
80-233, 11/12 Narutowicza Street, Gdansk, Poland
Email: pczarnul@eti.pg.gda.pl

***Abstract*—The paper presents design of a distributed system for measurements and control of a smart home including temperatures, light, fire danger, health problems of inhabitants such as increased body temperature, a person falling etc. This is done by integration of mobile devices and standards, distributed service based middleware BeesyCluster and a workflow management system. Mobile devices are used to measure the parameters and are coupled with computers for remote access. The latter is made possible by exposing services in the distributed middleware. Such services can then be integrated into complex workflow applications for constant monitoring and analysis of input data, instant feedback such as turning off lights or heating or notification of medical emergency. Interfaces are defined for particular system components and implementation hints for measurements using the Java Mobile Sensor JSR 256 API are provided.**

I. INTRODUCTION

Widespread availability of the Internet has made it possible to build hybrid distributed systems that couple various features of particular types of systems. The leading architectures and types of systems include nowadays:

1) cloud computing [1] that allows outsourcing of infrastructures (IaaS), platforms (PaaS) and software (SaaS),
2) grid computing [2] for controlled resource sharing especially referring to computational and storage resources,
3) high performance computing (HPC) [3] thanks to wide availability of the following:
    - multi-core processors with increasing numbers of cores,
    - high performance and multi-core GPUs allowing general purpose processing on GPUs (GPGPU).
4) extremely wide availability and usage of mobile devices, especially smartphones equipped with more and more sensors of various types.

The contribution of this work is proposal of a distributed system for remote measurements and control of various parameters of a smart home and people remotely. This is possible by a new integrated approach using mobile devices for measurements, high performance clusters for analysis, servers of various forces for notification and a workflow management system for definition and execution of processing flow and supporting security.

978-1-4673-5637-4/13/$31.00 © 2013 IEEE

II. RELATED WORK

A. Mobile Devices for Measurements

Modern smartphones are now equipped with a wide range of sensors that can be very useful in a variety of practical applications. The sensors include[1]:

1) ambient light sensor (ALS),
2) proximity sensor,
3) Global Positioning Sensor (GPS),
4) accelerometer,
5) compass,
6) gyroscope.

More and more types of sensors have just or are predicted to be provided in the latest mobile devices including[2]:

1) altimeter,
2) temperature, humidity sensors,
3) heart monitor,
4) perspiration sensor,
5) asthma device with a connection to a smartphone etc.

These sensors will provide even greater possibilities to use in smart homes and to monitor states and mood of patients, including prediction of danger due to high heart rate, high body temperature, too high altitude for the particular person etc. So far mobile technologies have been used to assist in data collection for e.g. measures of physical activity [4].

The Visibility app for Android smartphones developed at University of Southern California allows to measure air pollution using a mobile phone[3] [5].

[1] http://mobiledeviceinsight.com/2011/12/sensors-in-smartphones/
[2] http://mobihealthnews.com/11006/which-sensors-are-coming-to-your-next-smartphone/
[3] http://www.treehugger.com/clean-technology/android-app-measures-air-pollution-using-cell-phones-camera.html

Paper [6] deals with noise monitoring and mapping of noise in neighborhood using GPS-equipped mobile phones.

In many cases processing of such data that can be acquired using mobile technologies requires high performance computing hardware and parallelization of software [7].

B. Workflow Applications and Existing Approaches

Modeling complex processing as workflow applications has a wide coverage in the literature. Traditionally, workflow applications were defined for business [8], [9] and scientific [10] applications.

The workflow application is usually modeled as an acyclic directed graph [11], [10], [12] in which nodes correspond to tasks t_i being executed while the edges denote the order of tasks. For each workflow task t_i there is a set of services s_{ij} (j-th service for task t_i) capable of executing the given task with certain QoS parameters such as cost, execution time, reliability etc. The optimization problem requires assignment of $t_i \rightarrow (s_{ij}, time_{ij})$ i.e. a particular service executed at a particular moment in time ($time_{ij}$) so that a global QoS criterion is optimized such as minimization of the workflow execution time with a bound on the total cost of services or vice versa [13], [12].

A taxonomy of workflow applications is presented in [14] with possibilities to extend the model shown in [15].

In ubiquitous computing, contrary to the traditional workflow model, processing depends on the context [16], [17]. FollowMe as a platform allows to define workflows with Compact Process Definition Language with running using event triggering [18]. In the literature, there are also workflow applications used for the smart home. Paper [19] proposes a system that provides services to the user based on the current context information and uses Petri nets to model workflows. This is discussed in the context of application in a smart home. Furthermore, a new context-aware workflow language and a system for ubiquitous computing in the smart home context is proposed in [20]. The context information is specified when transition in the workflow occurs. Services based on context information can be provided. Paper [21] focuses on communication between heterogeneous appliances in a smart home and uses an information appliance interface definition language (IAIDL) and event-driven workflow processing.

There are Ambient Assisted Living (AAL) solutions available that deal with smart spaces oriented around human users. Smart devices within such environment gather information and act collectively within a defined scenario. Such research objectives are realized within project universAAL: an Open Platform and Reference Specification for Building AAL Systems[4]. It is interesting that such spaces can be managed remotely. Paper [22] discusses a multi-level model for development of context-aware AAL applications by definition of basic functions at one level and adaptation at the next level. State-of-the-art implementation concepts for AAL applications are discussed in [23].

[4] http://www.universaal.org

C. Motivations

The following factors motivated design of the proposed system including needs for:

1) easy installation and reconfiguration of measurement devices in any place of one or more smart homes,
2) monitoring both parameters of homes as well as inhabitants,
3) processing and analysis of the measured data including high performance processing capabilities e.g. for detection of persons who have broken into a house or apartment,
4) easy invocation of services that notify respective forces such as the police, fire stations or medical assistance about dangerous situations,
5) handling many smart homes and persons that belong to one owner,
6) ability to define and run various measurement and reaction scenarios for various owners at the same time,
7) support security mechanisms for the proposed solution,
8) reuse and adaptation of available workflow management solutions and middleware for service based distributed systems.

III. DESIGN OF THE PROPOSED SYSTEM

The goals stated in section II-C can be addressed in a distributed system proposed in this work through integration of ready-to-use components into one system for measurement, analysis and notification:

1) a workflow management system for distributed service-based systems including both business and scientific services developed by the author before [24],
2) mobile devices such as smartphones which are equipped with several sensors able to measure parameters wherever the device is left.
3) high performance computing (HPC) systems able to run complex processing of the measured data e.g. by a parallel MPI application.

A. Architecture

The architecture of the proposed system is shown in Figure 1. The following layers are distinguished from bottom to top:

1) mobile devices used for measurement of parameters such as temperature, light, visibility, noise, acceleration etc. The important advantage of using mobile devices are as follows:
   - can be flexibly put in any place and the location can be easily changed,
   - most of the latest mobile phones are equipped with a variety of sensors,
   - the mobile operating system does not affect the working of the system as the interface masks the operating system.
2) machines exposing services, two types of machines can be distinguished at this level:

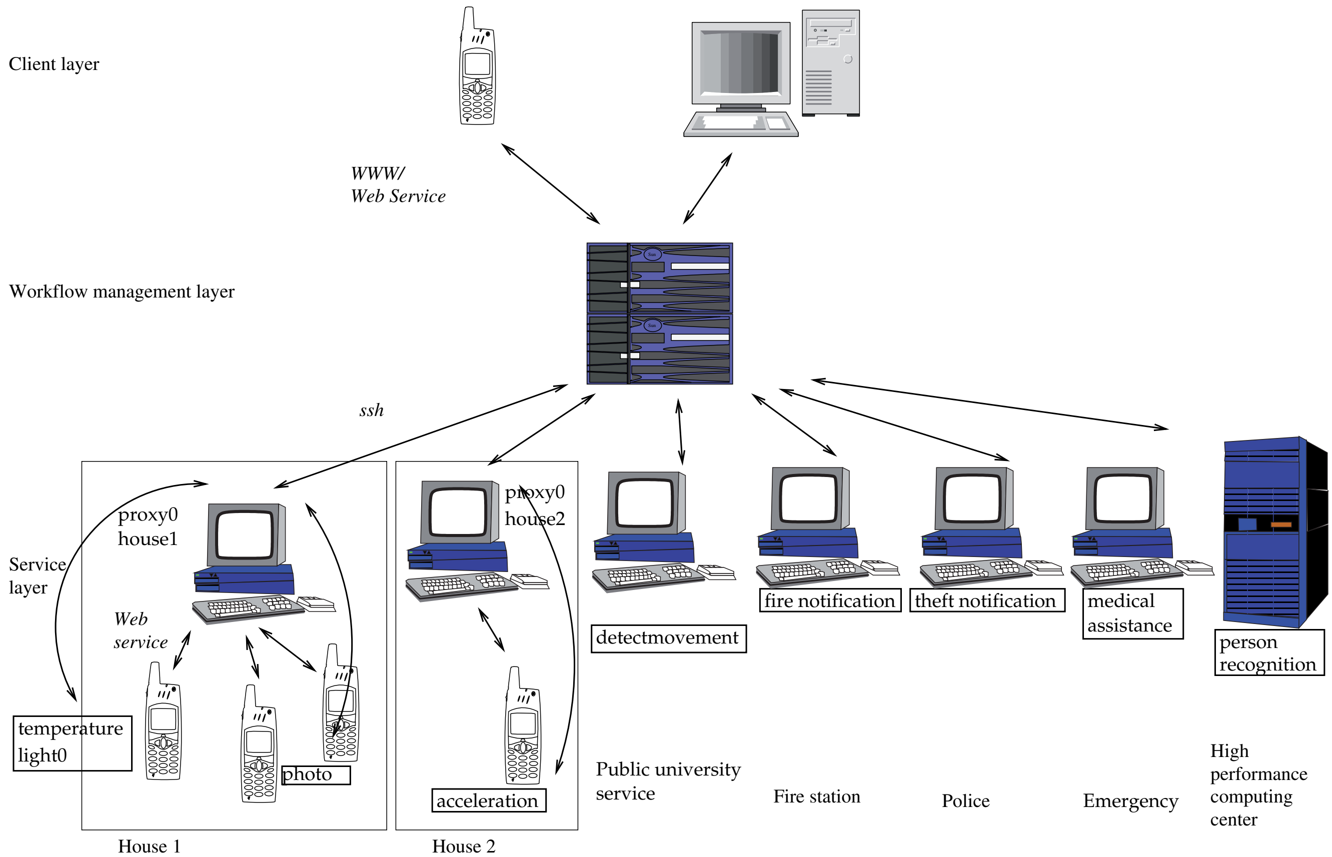

Fig. 1. Architecture of the proposed system

a) computers acting as proxies that allow access to one or more mobile devices. The proxies can be accessed by one or more clients by exposing services. Obviously, access to the proxies is secured to authorized parties with encryption of the data sent through the proxies.
b) servers exposing other ready-to-use services such as services for analysis of data or reaction to data exceeding various thresholds.

3) a workflow management system that models processing of the data that has been measured and launches services in response to values outside desired ranges. In fact, several workflow applications can be activated and run in parallel. For instance:

a) different workflow applications launched by various users to measure and control their smart homes. Various sensors and thresholds could be used in those workflows.
b) monitoring of several houses in one workflow. For instance, the workflow can monitor the temperature, light and if there are burglars inside a summer house and whether a person having a smartphone in a packet has not fallen down in his or her apartment.

B. Security

Security is a very important aspect in a distributed system, especially when accessing one's own home and its facilities from outside. The solution is secured at various levels in the following way:

1) communication between the client and BeesyCluster is secured using:
   a) passing login and password,
   b) encryption of communication (both WWW and Web Services) using HTTPS,
2) BeesyCluster stores access credentials to the servers on which it invokes services in a database. These fields are encrypted and inaccessible other than from the core components of the system. Depending on the previously verified and encrypted session of the user in BeesyCluster, the user is granted access to particular services. Invocation of the services is secured by SSH.
3) within the smart phone, the devices share the same LAN which is separated from the outside world using a NAT with proper port forwarding to the proxy computers. For the other servers, ports for the SSH need to be open with the other ones controlled by a firewall.

C. Services

Essentially, several types of services can be used in one workflow application:

1) measurement service – these are services that consist of two parts:
   a) measurement code implemented on a mobile device using the sensors such as light, temperature, acceleration etc.
   b) the measured data is then passed to a proxy server which exposes a service to fetch this data from the outside world in a secure way in the BeesyCluster middleware.
2) analysis services – the use of a workflow management system allows to incorporate services installed on both commodity servers and HPC systems. For instance, detection of a thief from a clip recorded on a mobile device can be performed by a parallel MPI program on a cluster. The analysis service can also apply certain filters on the measured data to eliminate short-term peaks or measurement errors.
3) notification services – to inform respective forces if thresholds in the measured values have been exceeded.

D. Components and Interfaces

The interesting contribution of this work is definition of interfaces of particular components that leaves the actual implementation to the programmer. This has the benefit of using the technology of choice preferred by the programmer. For instance, access to sensors on mobile devices can be performed using APIs specific for a particular mobile operating system or using e.g. Java Mobile Edition defined within JSR 256. The programmer needs to implement selected methods of the proposed API.

1) Mobile Service: The following APIs are defined in interface `MobileInterface` with the methods to be provided by the programmer being underlined:

1) <u>`bool SendCondition(MeasuredParameter [][])`</u> – checks on the mobile device if the data should be sent to the proxy. The array passes all measured channels as well as the history of measured values. For instance, for measurement of a falling person, successive values of acceleration in X, Y, Z dimensions must be checked. If the condition is met, then `true` will be returned. On the other hand, sometimes preprocessing of the input data is not possible on the mobile device. For instance, detection of a person, potentially a thief, in a frame is not a trivial task. This method may simply return true and rely on analysis services on an HPC system to process the content.
2) <u>`int SentWindowLength()`</u> – defines the width of the window for which results need to be sent including the current measured value. This applies to all the measured channels.
3) <u>`long MeasureInterval()`</u> – defines the interval within which measurements are performed,
4) <u>`long SendInterval()`</u> – defines how frequently the measured data is sent to the proxy server,
5) <u>`String ProxyWebServiceURL()`</u> – defines the address of the proxy Web Service,
6) <u>`String ServiceUniqueName()`</u> – defines the unique name of the measurement service.

2) Proxy: For the proxy server, the following methods of a Web Service are defined to be invoked by the measurement service:

1) `String Login(String login,String password)` – a token is returned that identifies a session and is used to submit data,
2) `String RegisterMobileService(String serviceName, String login, String password)` – performs automatic login and returns a token,
3) `long SubmitMeasurement(MeasuredParameter [][],String token)` – submits the data to the proxy. The proxy can define the time until when the next submission should occur. If not provided (returned value equal to 0), the send interval defined in the measurement service can be used.

Obviously, the proxy needs to store data in a database with values for the registered services. Apart from that, an API for the workflow management system must be provided. To that system, each registered mobile service is visible as a service in the BeesyCluster middleware which is described in section III-D3.

3) BeesyCluster and the Workflow Application: BeesyCluster[5] [25] is a middleware that allows access to distributed resources such as: high performance computing systems, commodity servers as well as workstations. It allows single sign-on and uniform access using WWW and Web Services to one or more user accounts on the distributed resources that have been registered in BeesyCluster before. Functions that the user can perform include management of files and directories including copying to and between the accounts, creating and editing files, compilation, running either sequential or parallel applications, the latter run on high performance computing systems. In the latter case, queuing systems such as PBS or LSF used by these systems are hidden to the user and running the application resembles running a standard application.

Apart from many other BeesyCluster functions, it is important for this work that applications available on such accounts can be made available as services. Among others, each service has the following parameters:

1) name,
2) location,
3) execution time,
4) cost,
5) time when it is to be run,
6) specification of output files e.g. extensions.

Furthermore, services can be used in workflow applications in the workflow management subsystem developed before [24], [26]. Since BeesyCluster can have access to a variety of systems as indicated above, the application of the workflow depends only on the combination of the services used.

[5] `https://beesycluster.eti.pg.gda.pl:10030/ek/AS_LogIn`

The workflow model used in BeesyCluster is as follows. The workflow application is modeled as an acyclic directed graph (DAG) in which nodes denote tasks to be executed while edges denote time dependencies between tasks and flow of data between successive tasks. For each task t_i of the workflow there is a set of services S_i containing services s_{ij} each of which is capable of executing the task. The services have parameters such as execution time t_{ij} and cost c_{ij} for a particular service s_{ij}. Each of the task is supposed to process data of size d_i which can be set in advance. Output data from a task is passed to successive tasks. The data can be partitioned for parallel processing by successive paths or same data can be passed to different tasks.

There are the following steps in workflow management:

1) workflow definition with the workflow graph and assignment of services capable of executing the particular tasks. Furthermore, the goal for selection of services needs to be set e.g. for selection of such services so that the workflow execution time is minimized with a bound on the cost of selected services $\sum_{i:s_{ij}\ selected\ for\ t_i} c_{ij} d_i$ where s_{ij} is the service selected for execution of task t_i,
2) optimization of workflow scheduling i.e. assignment of $t_i \rightarrow (s_{ij}, time_{s_{ij}})$ i.e. assignment of a particular service being executed for task t_i at moment $time_{s_{ij}}$. One of several scheduling algorithms can be used for this purpose. In general the problem with the optimization goal defined above is NP-hard which makes it necessary to use good heuristic algorithms. Examples of such algorithms include:
    - Integer Linear Programming based approach,
    - genetic algorithms,
    - divide-and-conquer where for a subproblem an optimal or another heuristic algorithm is used,
    - GAIN/LOSS.
3) workflow execution and monitoring [24], [26].

Obviously, it is also possible to assign one service per task which makes the optimization part redundant and the workflow is then called concrete rather than abstract that needs to be optimized.

For the purpose of the application considered in this paper, the workflow is constructed as follows:

1) SERVICES: BeesyCluster services are defined on two types of distributed resources:
    a) proxy servers in the smart home – these are services required to fetch values measured on mobile devices and possibly to control some external devices such as lights, heating etc.
    b) external resources such as a high performance computing (HPC) cluster used for analysis of the data sent from the house and on servers of various public forces such as: fire station to notify about fire, police to notify about potential theft and emergency in case some person requires immediate help.
2) QoS and FAILURE PROOFING: For some tasks, there can be just one service such as fetching the temperature in the house or notification about potential theft after recognition of a person inside a supposedly empty house or apartment. In this respect this makes the workflow concrete i.e. executes services one after another. However, it may be desirable to define more than one capable service for a task due to the following reasons:
    a) quality of service - e.g. it may be more desirable to contact directly the closest of two or more available fire or police stations.
    b) to make use of the failure proof solution within the BeesyCluster workflow management system: if one out of two or more capable services has failed or has become unavailable, then the system automatically reschedules the workflow and picks up the second best service [24].
3) DATA and STREAMING MODE: In the BeesyCluster workflow management module, each service assigned to a task processes data. The size of this data reflects the number of files processed by the service. It is up to the service how the file and its content is read and used. Similarly, the service is supposed to produce output data in the form of files (can have a special extension which is marked in the service description). The output data is automatically moved (if necessary across the network) to successive services. Furthermore, the system can work in two modes:
    - synchronous – where the service awaits for all input data before processing,
    - streaming – where the service is invoked for each of the data file individually as soon as it comes.

   The streaming mode is invoked in the workflow application proposed in this paper. That means that data files sent through a workflow path periodically act as triggers for successive services. This can be configured at the workflow application definition level as indicated in section IV.

In line with that, the workflow application shown in Figure 2 is proposed that does the following in parallel:

1) measurement of temperature and light in the remote smart home. Service `temperaturelight` fetches the values from the smart home that are periodically provided by the mobile device and cached on the proxy server. Successive invocation of service `temperaturelight` is done by sending data files acting as triggers for successive invocations of this service. If the conditions for fire are met a file with a specific extension is produced – the same as marked as output in the service description. It is then passed to service `firenotification` to trigger forces at the fire station.
2) measurement of successive values from the acceleration sensor mounted in a cell phone in a pocket of the person in the smart home. Depending on the setting `SentWindowLength()` in the mobile service, a certain window with the X, Y and Z channels are passed to the proxy. This data is then passed to service `detectmovement` installed on a powerful

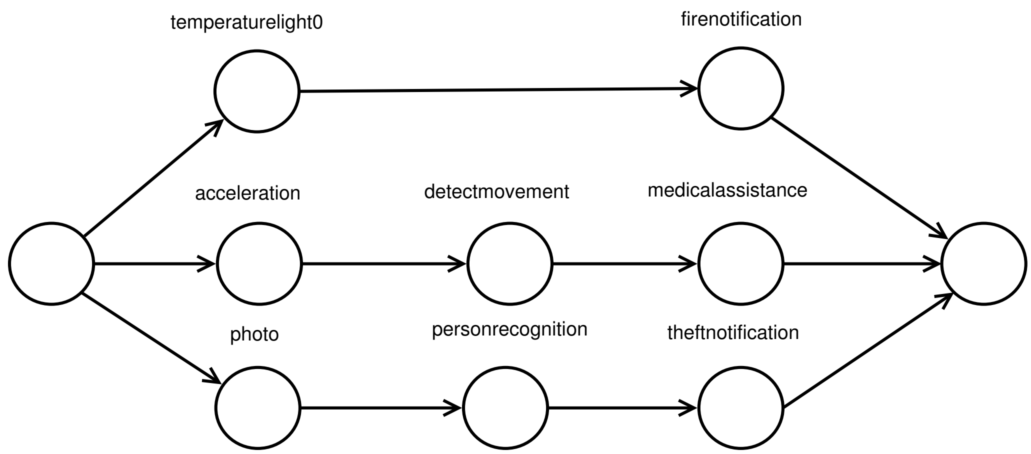

Fig. 2. Proposed workflow application

HPC system. If the pattern that denotes a falling and not moving person is detected, a file with a special extension is produced which triggers service `medicalassistance` of the respective force.

3) taking a photo by a camera in a mobile device (service `photo`) and passing to service `personrecognition` on a HPC system. If a person is detected in a supposedly empty house then service `theftnotification` is invoked in a police department. The latter is invoked by passing an output file with a specific extension produced by service `personrecognition`. This is the extension marked as the type of output file for that service as only such files are passed further in the workflow.

The way the system works is shown in the interaction diagram in Figure 3. Both registration of the mobile code in the proxies and subsequent execution of the workflow by the BeesyCluster workflow management system according to the workflow application from Figure 2 are shown.

IV. SAMPLE IMPLEMENTATION

This section gives recommendations and details on how particular system components need to be implemented.

Implementation requires coding and definition of the following:

1) underlined methods listed in section III-D,
2) setting up an account in BeesyCluster,
3) publishing measurement, analysis and notification services from accounts on proxy servers, HPC systems or servers in institutions the user has access to [25]. Alternatively, some of such services may have been published and made available to the user by others,
4) definition of and launching the workflow application.

As an example, the pseudocode shown in Figure 4 using Java Mobile Edition could be used for measurement of acceleration on a mobile device.

Furthermore, the workflow application from Figure 2 can be defined within BeesyCluster which is shown in Figure 5.

```
class MobileInterfaceImpl implements MobileInterface {
  String deviceId;
  // constructor
  ...
  bool SendCondition(MeasuredParameter [][] mp) {
    return true; // send out all values for analysis on a
        cluster
  }
  int SentWindowLength() {
    return <wlength>;
  }
  long MeasureInterval() {
    return <minterval>;
  }
  long SendInterval() {
    return <sinterval>;
  }
  String ProxyWebServiceURL() {
    return "Mobile service "+deviceId;
  }
}
class AccelerationMobile implements DataListener {
  SensorInfo sensor;
  SensorConnection connection;
  double axisX, axisY, axisZ;

  MobileInterfaceImpl mii=MobileInterfaceImpl();
  MeasuredParameter [][] mp=null;

  public AccelerationMobile() {
    initiate mp ...
//find the acceleration sensor
    SensorInfo sensors[] =SensorManager.findSensors(
      "acceleration", SensorInfo.CONTEXT_TYPE_DEVICE);
    sensor = sensors[0];
      try {
        connection = (SensorConnection) Connector.open(
            sensor.getUrl());
        connection.setDataListener(this, mii.SentWindowLength
            (),);
      } catch (IOException ex) {
        ...
      }
    } }
  public void dataReceived(SensorConnection con, Data[] data
      , boolean b) {
    copy the data from data to mp;
    // X <- data[0].getDoubleValues()[0];
    // Y <- data[1].getDoubleValues()[0];
    // Z <- data[2].getDoubleValues()[0];
  }
}
```

Fig. 4. Sample implementation for acceleration

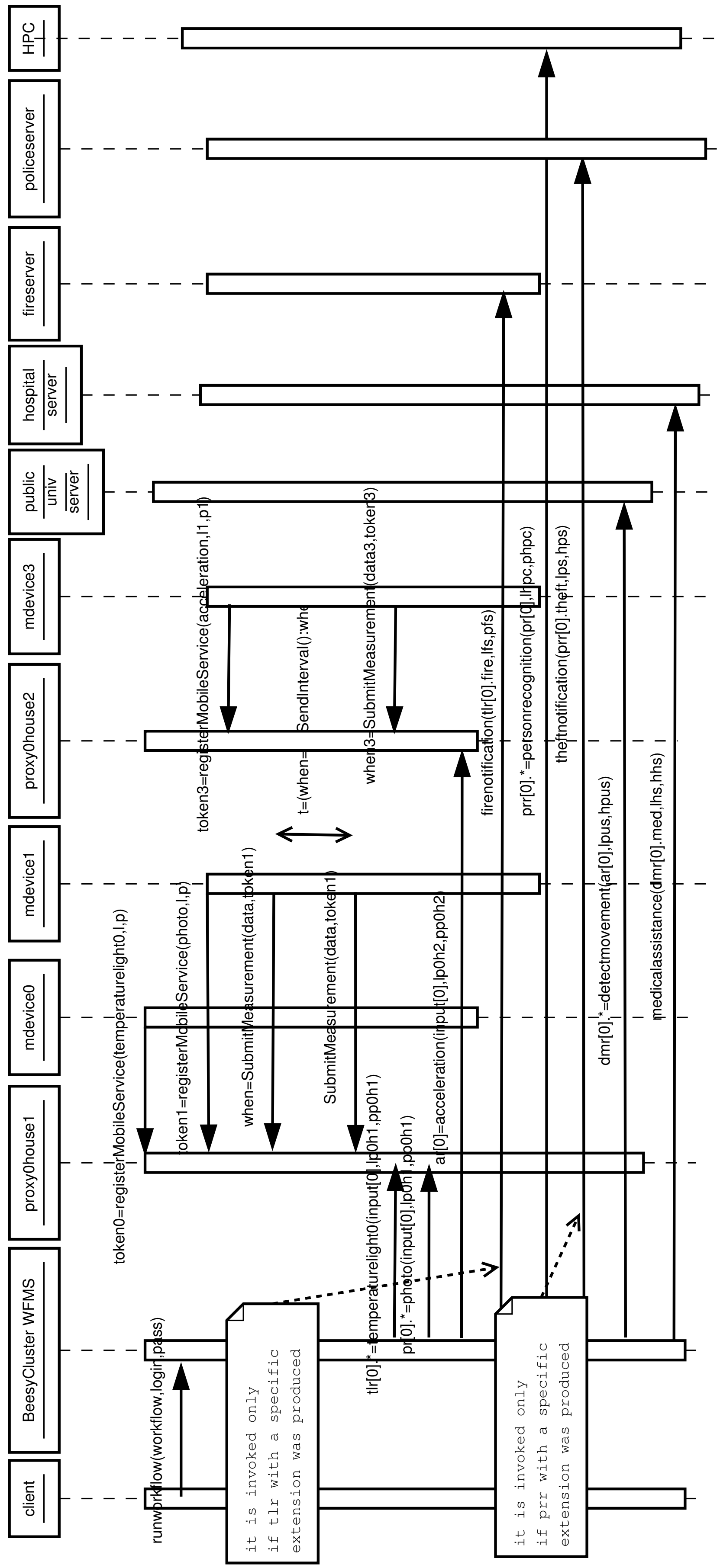

Fig. 3. Main interaction in the proposed system

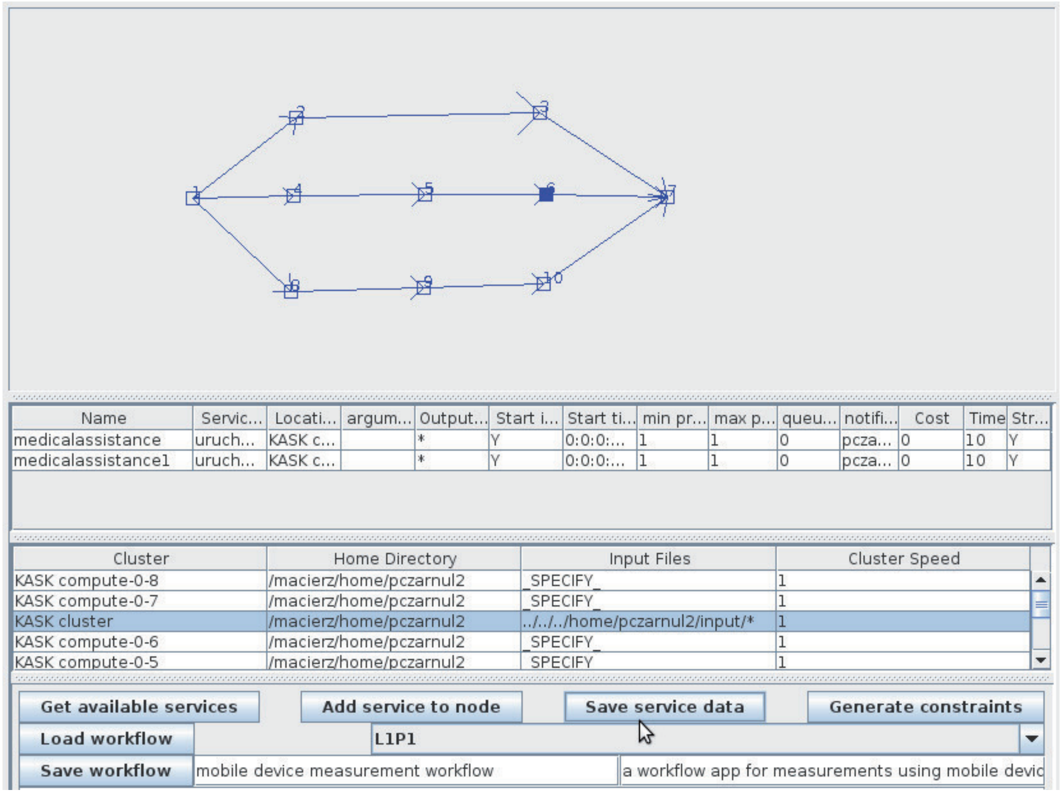

Fig. 5. Proposed workflow application in the BeesyCluster WfMS

BeesyCluster offers workflow management panels for launching and monitoring previously run workflow applications [24].

V. CONCLUSIONS

The paper presented design of a distributed system that integrates mobile devices for measurement of parameters and monitoring people in a smart home, high performance computing computers for processing of data as well as a workflow management system for definition of data and control flow as well as its execution. The paper proposed interfaces for particular components of the system and implementation hints using Java Mobile and Sensor API JSR 256 for mobile devices. It has also been shown how the real BeesyCluster middleware and its workflow management system can be used to model the actual workflow.

REFERENCES

[1] Ahson, S.A., Ilyas, M., eds.: Cloud Computing and Software Services: Theory and Techniques. CRC Press (2011) ISBN 978-1-4398-0315-8.

[2] Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. In: Open Grid Service Infrastructure WG. (2002) Global Grid Forum, `http://www.globus.org/research/papers/ogsa.pdf`.

[3] Buyya, R., ed.: High Performance Cluster Computing, Programming and Applications. Prentice Hall (1999)

[4] Bexelius, C., Lf, M., Sandin, S., Lagerros, Y.T., Forsum, E., Litton, J.: Measures of physical activity using cell phones: Validation using criterion methods. J Med Internet Res **12** (2010) doi: 10.2196/jmir.1298, PMID: 20118036.

[5] Poduri, S., Nimkar, A., Sukhatme, G.: Visibility monitoring using mobile phones (2009) `http://robotics.usc.edu/mobilesensing/visibility/MobileAirQualitySensing.pdf`.

[6] Maisonneuve, N., Stevens, M., Steels, L.: Measure and map noise pollution with your mobile phone. In: Proceedings of the DIY for CHI workshop held on April 5, 2009 at CHI 2009, the 27th Annual CHI Conference on Human Factors in Computing Systems, Boston, MA, USA (2009) 78–82

[7] Czarnul, P., Grzeda, K.: Parallelization of electrophysiological phenomena in myocardium on large 32 & 64-bit linux clusters. In Springer-Verlag, ed.: Proceedings of Euro PVM/MPI 2004, 11th European PVM/MPI Users' Group Meeting. Volume LNCS 3241., Budapest, Hungary (2004) 234–241

[8] Canfora, G., Penta, M.D., Esposito, R., Villani, M.: (A Lightweight Approach for QoS-Aware Service Composition) ICSOC 2004 forum paper, IBM Technical Report Draft.

[9] Canfora, G., Penta, M.D., Esposito, R., Villani, M.: Qos-aware replanning of composite web services. In: Procs. of 2005 IEEE International Conference on Web Services. Volume 1., Res. Centre on Software Technol., Sannio Univ., Italy (2005) 121–129

[10] Yu, J., Buyya, R.: Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. Scientific Programming Journal **14** (2006) 217–230 IOS Press, Amsterdam.

[11] Yingchun, Li, X., Sun, C.: Cost-effective heuristics for workflow scheduling in grid computing economy. In: GCC '07: Proceedings of the Sixth International Conference on Grid and Cooperative Computing, Washington, DC, USA, IEEE Computer Society (2007) 322–329

[12] Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., Sheng, Q.: Quality driven web services composition. In: Proceedings of WWW 2003, Budapest, Hungary (2003)

[13] Yu, J., Buyya, R., Tham, C.K.: Cost-based scheduling of workflow applications on utility grids. In: Proceedings of the 1st IEEE International Conference on e-Science and Grid Computing (e-Science 2005), IEEE CS Press, Melbourne, Australia (2005)

[14] Yu, J., Buyya, R.: A taxonomy of workflow management systems for grid computing. Journal of Grid Computing **3** (2005) 171–200

[15] Wieczorek, M., Hoheisel, A., Prodan, R.: Towards a general model of the multi-criteria workflow scheduling on the grid. Future Generation Computer Systems **25** (2009) 237 – 256

[16] Ben Mokhtar, S., Fournier, D., Georgantas, N., Issarny, V.: Context-Aware Service Composition in Pervasive Computing Environments. In: Rapid Integration of Software Engineering Techniques, Second International Workshop : RISE 2005, Heraklion, Crete Greece (2006) 129–144

[17] Han, J., Kim, E., Choi, J.: Workflow language based on web services for autonomic services in ubiquitous computing. In: Proceedings of International Conference on Artificial Reality and Telexistence, ICAT, Coex, Korea (2004)

[18] Li, J., Bu, Y., Chen, S., Tao, X., Lu, J.: FollowMe: on research of pluggable infrastructure for context-awareness. In: Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference on. Volume 1. (2006)

[19] Xing, Z., Hong, Z., Yulong, L.: A petri-net based context-aware workflow system for smart home. In: Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW), 2012 IEEE 26th International. (2012) 2336 –2342

[20] Cho, Y., Choi, J., Choi, J.: A context-aware workflow system for a smart home. In: Convergence Information Technology, 2007. International Conference on. (2007) 95 –100

[21] Cong, Z., Hong, L., Wen-bo, L.: Application of workflow technology in smart home based on expert system. Computer Technology and Development (2008) 205–208 Issue 9, 1673-629X.

[22] Segarra, M., Andre, F.: Building a context-aware ambient assisted living application using a self-adaptive distributed model. In: Autonomic and Autonomous Systems, 2009. ICAS '09. Fifth International Conference on. (2009) 40–44

[23] Rcker, C.: Designing ambient assisted living applications: An overview over state-of-the-art implementation concepts. In: 2011 International Conference on Modeling, Simulation and Control. Volume 10., Singapore, IACSIT Press (2011)

[24] Czarnul, P.: Modeling, run-time optimization and execution of distributed workflow applications in the jee-based beesycluster environment. The Journal of Supercomputing **63** (2013) 46–71 DOI: 10.1007/s11227-010-0499-7.

[25] Czarnul, P., Bajor, M., Fraczak, M., Banaszczyk, A., Fiszer, M., Ramczykowska, K.: Remote task submission and publishing in beesycluster : Security and efficiency of web service interface. In Springer-Verlag, ed.: Proc. of PPAM 2005. Volume LNCS 3911., Poland (2005)

[26] Czarnul, P.: A model, design, and implementation of an efficient multithreaded workflow execution engine with data streaming, caching, and storage constraints. The Journal of Supercomputing **63** (2013) 919–945 DOI: 10.1007/s11227-012-0837-z.