International Journal on Advances in Intelligent Systems

2009 vol. 2 nr. 2&3

The *International Journal on Advances in Intelligent Systems* is Published by IARIA.
ISSN: 1942-2679
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Intelligent Systems
Volume 2, Numbers 2&3, 2009

- Thomas C. Schmidt, University of Applied Sciences – Hamburg, Germany
- Karolj Skala, Rudjer Bokovic Institute - Zagreb, Croatia
- Chieh-yih Wan, Intel Corporation, USA
- Hoo Chong Wei, Motorola Inc, Malaysia

Ubiquitous Systems and Technologies

- Matthias Bohmer, Munster University of Applied Sciences, Germany
- Dominic Greenwood, Whitestein Technologies AG, Switzerland
- Arthur Herzog, Technische Universitat Darmstadt, Germany
- Reinhard Klemm, Avaya Labs Research-Basking Ridge, USA
- Vladimir Stantchev, Berlin Institute of Technology, Germany
- Said Tazi, LAAS-CNRS, Universite Toulouse 1, France

Advanced Computing

- Dumitru Dan Burdescu, University of Craiova, Romania
- Simon G. Fabri, University of Malta – Msida, Malta
- Matthieu Geist, Supelec / ArcelorMittal, France
- Jameleddine Hassine, Cisco Systems, Inc., Canada
- Sascha Opletal, Universitat Stuttgart, Germany
- Flavio Oquendo, European University of Brittany - UBS/VALORIA, France
- Meikel Poess, Oracle, USA
- Said Tazi, LAAS-CNRS, Universite de Toulouse / Universite Toulouse1, France
- Antonios Tsourdos, Cranfield University/Defence Academy of the United Kingdom, UK

Centric Systems and Technologies

- Razvan Andonie, Central Washington University - Ellensburg, USA / Transylvania University of Brasov, Romania
- Kong Cheng, Telcordia Research, USA
- Vitaly Klyuev, University of Aizu, Japan
- Josef Noll, ConnectedLife@UNIK / UiO- Kjeller, Norway
- Willy Picard, The Poznan University of Economics, Poland
- Roman Y. Shtykh, Waseda University, Japan
- Weilian Su, Naval Postgraduate School - Monterey, USA

GeoInformation and Web Services

- Christophe Claramunt, Naval Academy Research Institute, France
- Wu Chou, Avaya Labs Fellow, AVAYA, USA
- Suzana Dragicevic, Simon Fraser University, Canada
- Dumitru Roman, Semantic Technology Institute Innsbruck, Austria
- Emmanuel Stefanakis, Harokopio University, Greece

Semantic Processing

- Marsal Gavalda, Nexidia Inc.-Atlanta, USA & CUIMPB-Barcelona, Spain
- Christian F. Hempelmann, RiverGlass Inc. - Champaign & Purdue University - West Lafayette, USA
- Josef Noll, ConnectedLife@UNIK / UiO- Kjeller, Norway
- Massimo Paolucci, DOCOMO Communications Laboratories Europe GmbH – Munich, Germany
- Tassilo Pellegrini, Semantic Web Company, Austria
- Antonio Maria Rinaldi, Universita di Napoli Federico II - Napoli Italy
- Dumitru Roman, University of Innsbruck, Austria
- Umberto Straccia, ISTI – CNR, Italy
- Rene Witte, Concordia University, Canada
- Peter Yeh, Accenture Technology Labs, USA
- Filip Zavoral, Charles University in Prague, Czech Republic

International Journal on Advances in Intelligent Systems
Volume 2, Numbers 2&3, 2009

CONTENTS

Integrating Web-based Sensor Information into Geospatial Mass-market Applications through OGC Web Processing Services

Theodor Foerster
International Institute for Geo-information Science and Earth Observation (ITC), Enschede, the Netherlands
foerster@itc.nl

Bastian Schaeffer
Institute for Geoinformatics, Muenster, Germany
schaeffer@uni-muenster.de

Simon Jirka
Institute for Geoinformatics, Muenster, Germany
jirka@uni-muenster.de

Johannes Brauner
Professorship of Geoinformation Systems, Technische Universität Dresden, Dresden, Germany
johannes.brauner@tu-dresden.de

Abstract

Sensor data are currently increasingly available on the web through Sensor Web technology. Sensors are an important asset in crisis management situations. However, to make decisions sensor information is required. This information is extracted using geoprocesses provided by for instance Web Processing Services. This paper describes an approach to provide sensor information to ordinary users by integrating sensor data and geoprocesses into mass-market applications. The applicability of the approach is demonstrated by a risk management scenario. The software presented has been developed within the Geoprocessing Community of the 52°North initiative and is available through an Open Source license.

Keywords: Geospatial mass-market applications, OGC WPS, Google Earth, Sensor Web.

1. Introduction

Sharing and accessing web-based geo-information (GI) is a key aspect in geospatial mass-market applications (such as Google Earth™ and Yahoo Maps™) and helps people answer spatially related questions. Currently, data services are organized in Spatial Data Infrastructures (SDIs) and allow anybody to access data on the web from anywhere at anytime. Those data services have matured within the last years, such as the suite of services published by the Sensor Web Enablement (SWE) initiative. However, sensor data available through the so-called Sensor Web often requires certain processing to answer a specific question. As most of the current data are available through web services, the processing should also be established on the web. For this reason, the Open Geospatial Consortium (OGC) released the Web Processing Service (WPS) interface specification [1] and realizes thereby a shift from services providing data towards services providing information. Additionally, Web Processing Services are promising as the availability of network (gigabit bandwidth) and processing capabilities (such as provided by multiple processor cores and advanced processing hardware) increases. In general, Web Processing Services provide a means to customize data offered by data services mostly located in SDIs.

The extraction of information from such web-based sensor data and its integration into current geospatial mass-market applications is promising to provide user-friendly access to up-to-date information and thereby helps users to answer their questions regarding a geospatial context. However, such integration has not been realized yet.

This paper analyzes the technological requirements of geospatial mass-market applications towards integrating sensor information by the means of web-based geoprocesses. Furthermore, it describes an approach to realize this integration. Finally, the paper demonstrates the applicability of the proposed approach by a fire threat use case. This use case demonstrates how ordinary users can access the most current information through their geospatial mass-market application and take actions accordingly. The described scenario addresses a forest fire assessment use case, which is one of four key areas of the OSIRIS project[1].

OSIRIS (Open architecture for Smart and Interoperable networks in Risk management based on In-situ Sensors) is a European integrated project within the Sixth Framework Programme, aligned with GMES (Global Monitoring for Environment and Security). The main goal of the project comprises of the design, development and testing of a service-oriented architecture for risk monitoring and crisis management. A special focus is put on web services for integrating in-situ sensors and sensor data as well as higher level user services. Furthermore, during the design of the OSIRIS architecture, open standards, especially those provided by the OGC, are used as a valuable input.

The tools implementing the presented approach have been developed within the Geoprocessing Community[2] of the 52°North initiative and are available through an Open Source license (GNU General Public License 2).

The remainder of the paper is structured as follows. First the key concepts applied in the presented approach will be described. Section 3 will present the benefits of providing sensor information by the means of web-based geoprocesses. Based on this, Section 4 will present the developed approach. The details about the implementation of the approach is described in Section 5. The developed approach then is applied to the risk management scenario in Section 6. The paper ends with a conclusion and elaborates on future work items.

This paper goes beyond the originally published work of [2] by reviewing work in the context of Sensor Web and providing insights into integrating Sensor Web and web-based geoprocessing.

2. Background

The term *geospatial mass-market application* is closely related to what has been called *Neogeography* [3,4] and *Volunteered Geographic Information* (VGI; [5]). Both concepts describe processes for creating, sharing and annotating geodata (e.g. locations) through web-based applications by the public and can be summarized under the term *Geoweb* [3]. There are several different software vendors active within this market, providing data, applications and services such as Google, Yahoo, Microsoft or NASA. One of the commonly used formats to exchange geospatial-related content within geospatial mass-market applications is *KML*. The following subsections will shortly introduce the KML standard, the WPS interface specification and the components of the Sensor Web.

2.1. The OGC KML standard

KML is widely used for geospatial mass-market applications such as Google Maps and Google Earth. Most lately, it became an official OGC standard [6]. KML is unique in the family of OGC data encodings, as it combines data encoding, styling and the special network facilities, which are called *NetworkLinks* and are also known as dynamic KML. These NetworkLinks are especially interesting in the context of web-based information, as they allow the dynamic integration of remote resources. Therefore, the content of a KML file might become dynamic and provide temporal data (e.g. from sensors). As NetworkLinks use URLs, KML is not only bound to file-based access, but can link any web service, as long as it operates via HTTP-GET and serves KML. NetworkLinks provide additional properties such as update, scheduling etc.

It has to be noted that NetworkLinks have already been integrated in GIS applications such as described by [7], but not for processing purposes.

2.2. OGC Web Processing Service

The WPS interface specification (OGC 2007) describes a standardized set of operations to publish and execute any type of geoprocess on the web. According to the WPS interface specification, a process is defined as an algorithm, calculation or model that operates on spatially referenced data.

In detail, the WPS specification describes three operations, which are all handled in a stateless manner: *GetCapabilities*, *DescribeProcess* and *Execute*. GetCapabilities is common to any type of OGC Web Service and returns service metadata. In case of a WPS it also returns a brief description of the processes offered by the specific WPS instance. To get further information about the hosted processes, the WPS is able to return the process metadata through the *DescribeProcess* operation. This operation provides the description of all parameters, which are required to run the process. Based on this information the client can perform the *Execute* operation upon the designated process. As any OGC Web Service, the WPS communicates through HTTP-GET and HTTP-POST using a messaging based on an OGC-specific XML-encoding.

[1] OSIRIS project website: www.osiris-fp6.eu.

[2] 52°North Geoprocessing Community website: www.52north.org/wps.

Besides that, the WPS interface specification describes mechanisms for asynchronous processing, processing of URL-referenced data and storing of process results. Especially the storing of process results is promising in the context of the presented approach. It allows to access process results server-side whenever required using a simple URL without re-initiating the process itself. Additionally, it is possible to request process results in a raw-data mode.

WPS implementations have already been successfully applied in several projects ranging from groundwater vulnerability analysis [8], bomb threat detection scenarios [9] and map generalization [10]. Additionally, an extensive discussion about the applicability of the WPS and its current drawbacks can be found in [11].

2.3 The Sensor Web

The activities of the OGC are centered on the idea of the so-called Geospatial Web [12]. This describes the aim to integrate geospatial data sources as well as processing capabilities in to the WWW. One aspect of the Geospatial Web, the integration of sensors and (potentially real time) data delivered by these sensors, is addressed by the Sensor Web Enablement Working Group [13].

The SWE Working Group has created a comprehensive framework of standards that aim at fulfilling the following objectives of the Sensor Web:

- Accessing sensor data (observations made by sensors)
- Tasking sensors (e.g. setting the parameters of a sensor)
- Alerting if user-defined alert conditions are matched by sensor measurements (e.g. alert if the water level at a certain location reaches a critical threshold)
- Accessing sensor parameters (e.g. information about the sensor configuration)
- Retrieving descriptions of sensors (sensor metadata)
- Discovering sensors and observations made by sensors.

The resulting framework of standards is divided into two parts (Figure 1): The SWE Information Model addresses the encoding of sensor data and metadata whereas the SWE Service Model consists of standards defining (web service) interfaces that offer the required functionality. In the remaining paragraphs of this subsection a short introduction into the SWE framework will be given to illustrate the architecture.

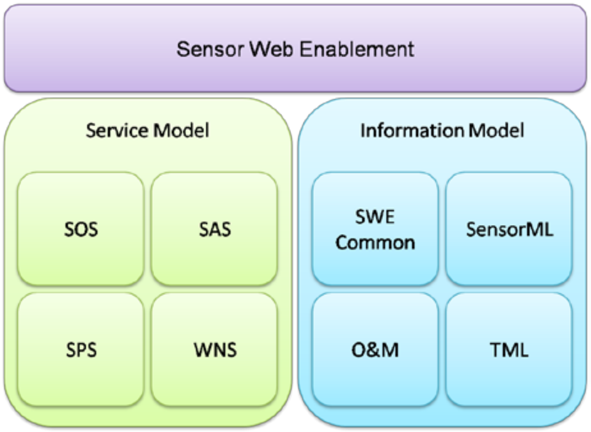


Figure 1: Overview of the SWE framework.

As previously explained the SWE Information Model comprises a set of standards defining the SWE data formats. These encodings provide means for exchanging sensor observations and metadata about the sensors in a well defined way. In detail, the following four standards are part of the SWE Information Model:

- SWE Common [14]: There are several basic building blocks (e.g. simple data types) that are used across the different SWE encodings. These elementary types are standardized within SWE Common.
- Observations and Measurements (O&M) [15,16]: O&M is the encoding for the data captured by sensors (= observations and measurements).
- Sensor Model Language (SensorML) [14]: Besides the data delivered by sensors it is also important to describe metadata about the sensor so that measurements can be interpreted, their quality can be assessed and previous processing steps can be documented. SensorML offers an encoding to provide this kind of metadata in a standardized way.
- Transducer Markup Language (TML) [17]: TML is capable of encoding sensor data as well as metadata. However it was developed as a further standard to provide a data format that is optimized to support data streaming.

Besides defining the data formats for exchanging sensor data and metadata there must also be a common approach is required for interfaces providing sensor related functionality. Such a set of interface standards is defined by the SWE Service Model. Four different standards belong to the SWE Service Model:

- Sensor Observation Service (SOS) [18]: The core functionality of the SOS is providing access to sensor data and metadata. It allows defining filter criteria to select the data on interest but also operations for inserting sensors and observations. Usually SOS instances return sensor data encoded in O&M and sensor metadata encoded in SensorML.
- Sensor Alert Service (SAS) [19]: Whereas the SOS is relying on a pull-based communication model, the SAS is able to push data to subscribers. These subscribers are able to define alert conditions in which they want to receive notifications.
- Sensor Planning Service (SPS) [20]: The SPS covers the issue of tasking sensors. This means that the SPS can be used for submitting tasks to sensors but also to manage these jobs (e.g. deleting, updating, and canceling).
- Web Notification Service (WNS) [21]: Finally, the WNS adds the capability of asynchronous communication to the SWE architecture. Thus, it can be seen as a helper service that can be used by the other SWE service (e.g. by the SAS for transmitting alerts).

3. Sensor Web and Web-based Geoprocesses

The integration of Sensor Web data and web-based geoprocesses (published as WPS) has been described for the application of smoke forecast models [22]. In this study, the authors distinguish between data processing services and data analysis services. Data processing services are considered to be placed in-situ on the actual sensor data service to reduce the amount of transferred data (tight coupling). The data analysis service combines data from distributed sources to create the desired information (loose coupling).

To integrate Sensor Web and web-based geoprocesses successfully, interoperability is crucial. The Sensor Web as described in Section 2.3, with its Information Model creates the foundation to encode the gathered data. The Service Model provides a means to deliver the encoded data in a standardized fashion. Both models build the starting point to transform the data into information.

As already discussed by [22], data processing can either be performed tightly coupled to the sensor (i.e. as a data processing service), or performed in a distributed fashion (data analysis service).

By tightly coupling the sensor with processing capabilities, such as in a simple case a temperature sensor only transmitting average values, the sensor device is able to perform the necessary calculations in-situ. This can be useful in several scenarios, especially for simple processes and low volumes of data. When it comes to processing large volumes of data and complex analysis functions of multiple data, the application of loosely-coupled and distributed services is more useful, since the processing power and storage power does not need to be located where the sensor is. Additionally, such web service-based analysis is reusable in different context with different sensor data. Limited power supply is also in many cases a crucial issue for sensor networks. Thus it is beneficial to process the data outside the sensor network and thereby to decrease the power consumption on the sensor web. Moreover, processing the data in a distributed fashion as a data analysis service turns into a requirement, when accessing multiple data sources is necessary.

By using WPS to generate information from Sensor Web data, there are basically two ways of communication:

- Data sent by value
- Data sent by reference.

Especially transmitting data by reference allows a WPS to obtain the latest available data. This is especially interesting to set up persistent service chains for analyzing latest sensor data. The service chain is set up once and performed whenever it is requested, analyzing latest data (as specified by the reference). Furthermore, the references can be used to implement caching strategies, as the WPS can identify data based on the reference without retrieving it again. However, there is always the problem of retrieving latest data and processing out-dated data (without knowing). Applying a specific caching strategy mostly depends on the actual application scenario. The main aim is to reduce the latency of the system for the end user [23].

When integrating Sensor Web and web-based geoprocesses to analyze real time sensor data the Sensor Observation Service seems to the suitable entry point to access the Sensor Web. The data provided by SOS is encoded as O&M and can be used easily within WPS via reference.

The following section describes the integration of these two services using a mass-market application. It also illustrates the idea of designing a service chain once and running it multiple times based on the latest sensor data.

4. Integrating Sensor Information into Geospatial Mass-market Applications

To integrate web-based sensor information, several requirements of the geospatial mass-market applications have to be met. The major requirement is that the communication pattern (REST architecture & KML encoding) of the geospatial mass-market applications does not have to be changed. Thereby, the sensor information becomes capable of being seamlessly integrated into such applications. This requirement is met by the WPS interface specification, as it allows the invocation of processes via HTTP-GET. Additionally, the WPS interface specification does not foresee any data encoding for its input and output parameters, thus KML is a valid format for the WPS. Finally, as the WPS is able to return process results as raw data without any WPS-specific message overhead it is highly applicable for the integration into geospatial mass-market applications.

The integration of sensor information into mass-market applications is possible through WPS interface due to the following aspects:

- It transforms sensor data into sensor information through geoprocesses
- It converts internally the sensor data from O&M into KML.

As the configuration of such processes is highly complex and not supported by current geospatial mass-market applications, this paper proposes a two-fold approach (Figure 2). At first, the selection and configuration of the process should be done by an expert user, utilizing an expert user interface, most-likely a Geographic Information System (GIS) such as the *User-friendly Desktop Internet GIS* (uDig[3]). At second, the user can integrate the pre-configured process into his geospatial mass-market application of choice. This is possible, as the WPS interface meets the requirements of geospatial mass-market applications, as explained in the previous paragraph. It is important to note, that the pre-configuration is not only necessary because of the lack of support for process configuration in geospatial mass-market applications, but is also highly applicable, as process configuration involves a lot of expert user knowledge. Thus the pre-configuration eases the integration of such information (extracted by geoprocesses) in mass-market applications for the user and is thereby not considered as a drawback.

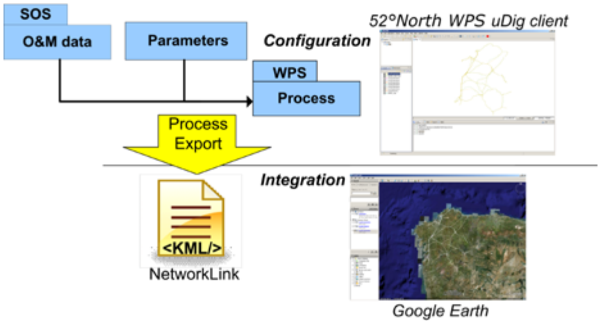


Figure 2: Approach to integrate sensor information in mass-market applications such as Google Earth.

For this study, the processes are configured through the 52°North WPS uDig client. This client application is extensively described in [24]. uDig has been enhanced to export these configured processes as KML, to integrate these geoprocesses and their results into geospatial mass-market applications. The export of the process from uDig to KML can be configured in two ways:

1. Export the KML file as a link to a stored process result. This is the *static* option, in which no process will be triggered when visualizing the KML file in Google Earth. This uses the store functionality of the WPS interface specification
2. Export the KML file as a link, which triggers the process on the WPS. This is the *dynamic* option and enables to trigger the process live, when incorporating the file in Google Earth. This allows one also to set a refresh rate to initiate the process on the server again. It is important to note, that in this case, the WPS process is triggered and if any WPS input data is defined as reference, the (updated) data is fetched and used as the basis for the calculation. This approach allows the processing of the latest available sensor data and thus visualizing the latest results in mainstream applications.

In both cases the files incorporate the links using the *NetworkLink* functionality of KML. Listing 1 shows the generated NetworkLink using the dynamic option in the KML export of uDig (option 2). The generated KML includes an Execute request via HTTP-GET to a spatial Buffer algorithm, which is also used in the scenario described in Section 6. The request references remote O&M data served by a SOS instance.

[3] uDig website: udig.refractions.net.

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://earth.google.com/kml/2.2">
  <Folder>
     <name>Buffered Features</name>
              <visibility>0</visibility>
              <open>0</open>
              <description>WPS Layer</description>
                <NetworkLink>
                     <name>WPS Layer</name>
                     …
                     <description>WPS Layer</description>
                     <refreshVisibility>0</refreshVisibility>
                    <Link>
<href>http://geoserver:8080/wps/WebProcessingService?request=ex
ecute&amp;service=WPS&amp;version=1.0.0&amp;Identifier=org.
n52.wps.server.algorithm.SimpleBufferAlgorithm&amp;DataInputs
=FEATURES=@mimeType=text/xml@href= http%3a%2f%2fv-
wupper%2f52nSOSv2_ArcSDE%2fsos%3fService%3dSOS%26Ver
sion%3d0.0.0%26Offering%3dwv.offering2%26observedProperty%
3dwv.phenomenon10%26responseFormat%3dtext%2fxml%3bsubty
pe%3d%2522om%2f0.0.0%2522@Schema=http://schemas.opengis.
net/om/1.0.0/om.xsd;DISTANCE=20&amp;RawDataOutput=BUFF
ERED_FEATURES@mimeType=application/vnd.google-
earth.kml%2Bxml@schema=http://www.opengis.net/kml/2.2</href>
              <refreshMode>onInterval</refreshMode>
              <refreshInterval>20</refreshInterval>
   …
</kml>
```

Listing 1: KML NetworkLink with a WPS-Execute request via HTTP-GET. The request references sensor data provided by SOS.

By supporting these two options (dynamic vs. static), the integration is well-scalable and applicable to scenarios requiring dynamic or even static process results. In case of integrating sensor data, dynamic process results might be more applicable to always obtain the latest sensor information.

It is important to note, that the client is able to perform and export single and chained processes. Chained processes are built up by using the output of a process as input for another process. This task of chaining is performed locally at the client side and does not involve any central processing engine. A more sophisticated approach in the context of web-based geoprocessing by the means of the Business Process Execution Language (BPEL) is described in [25].

The overall architecture with its different components (e.g. WPS and SOS) and clients (e.g. uDig and Google Earth) is depicted in Figure 3.

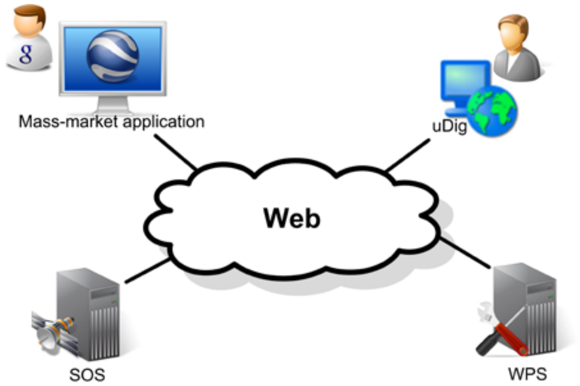


Figure 3: Overview of the involved components.

5. Implementation

This section presents the implementation of the components incorporated in the architecture. The described implementations are available as Open Source software through the 52°North initiative[4]. The implementations described in this section are all based on the Java programming language. The implemented Web Services (WPS and SOS) are deployed in servlet containers such as Tomcat.

5.1 52°North Initiative

As described above the architecture is based on the 52° North WPS and SOS framework.
52° North is an international research and development network which supports the process of advancing early research results to stable implementations that can be productively used in the geospatial domain. To strengthen this process, 52° North is shaped as an open membership initiative that brings together partners from research, industry and practical users.
The implementations published on the 52° North platform are available through a dual license model. This means that all software packages are published under the GNU General Public License 2 (GPL2) and concurrently under a commercial license.

5.2 52°North WPS Framework

The 52°North WPS framework is fully-compliant to the WPS interface specification version 1.0.0 and is developed and maintained by the 52°North Geoprocessing Community. The framework is based on a pluggable architecture, which is extensible regarding the designated processes and data handlers. It can be

[4] 52°North website: www.52north.org.

configured easily through a browser-based user interface and also integrates other geoprocessing functionality such as Sextante [26] and GRASS [27].

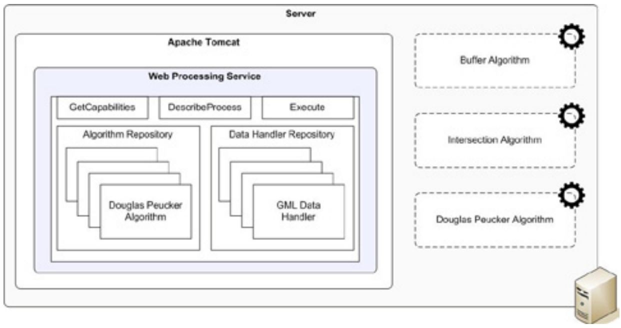


Figure 4: Architecture of 52°North WPS framework.

Besides the features as defined by the WPS interface specification, the framework supports workflow management [28] and grid computing [29]. These aspects are also subject to further research and have been identified as future activities of the Geoprocessing community.

For this study, the WPS has been enhanced to serve the sensor data (encoded as O&M) as KML. In particular, the data models are mapped internally by the WPS. This has been achieved in two steps. At first, the WPS requests the sensor data from the SOS and transforms the resulting O&M data into an internal data model, such as the Geotools feature model. The designated process is then performed on this internal data model. At second, the processed data are transformed into the external output data model, in this case KML. In both cases mappings are required between the O&M and the internal data model (e.g. Geotools feature model) and between the internal data model and KML.

5.3 52°North WPS uDig Client

The 52°North WPS uDig client is designed as a plug-in for uDig [25]. It is able to configure and perform remote functionality provided through the WPS interface. The specific process can be configured through a user-friendly wizard, which is generated on-the-fly, based on the input and output parameters of the designated process. The result of the performed process appears as a separate layer in uDig and can thereby be combined with other resources. As uDig is also able to integrate remote Web Services such as SOS, it is possible to send the data via reference. This allows the WPS to process directly remote resources.

Additionally, the 52°North WPS uDig client supports to export configured WPS processes as KML, which has been developed for the presented study.

5.4 52°North SOS Framework

The 52° North SOS framework is the foundation for this study to serve the sensor data in the presented architecture. It is fully-compliant to the OGC SOS standard version 1.0.0. Besides the mandatory operations of the core profile for accessing sensor data and metadata, it offers full support of the so-called transactional profile. This enables to insert sensors as well as their observations through a standardized web-based interface.

As the 52° North SOS framework relies on a modular design the adaptation and customization of the implementation is supported.

In the presented architecture the 52° North SOS framework serves the sensor data based on a PostgreSQL database (including the PostGIS extension).

6. Use Case Scenario

The scenario is applied to a risk management use case, in which in-situ-sensor data have to be analyzed for assessing a fictive fire threat in Northern Spain. The scenario and the involved services have been extensively presented in [24]. Currently, the services and data are taken from the ORCHESTRA project, which addresses a similar fire risk management scenario [30]. In a later stage of the project, data will be collected by sensors, as it is also the aim of the OSIRIS project. This section will focus on a modification and extension of the OSIRIS fire threat scenario, in which information has to be derived from real-time sensor data and the process results have to be disseminated to inform the public about the current situation.

Other examples of forest fire use cases are presented by [31,32]. For instance [31] present a web-based architecture, which is not based on standards and it is thereby not possible to adopt their architecture and approach to any similar use case. However, the presented approach (Section 4) is applicable to any other use case due to the interchangeable notion of web services. The chosen use case demonstrates the benefits of the approach and the resulting architecture.

According to the approach described in Section 4, the expert configures the process in the 52°North WPS uDig client. In particular the user configures a buffer of the sensor data to indicate potential fire threats and intersecting them with the road data. The buffer operation and the intersection operation are single processes hosted on a WPS. The road data has been additionally simplified to improve the process performance and to improve portrayal at smaller scales. Overall, this allows the expert to assess the parts of the road infrastructure which are at risk by a fire threat. The expert user exports the configured process as a KML file and links it on the national portal site. The citizen (i.e. ordinary user) is now able to visualize the latest analysis results in his/her geospatial mass-market application by loading the KML file from the portal site. He/she can inspect the latest data with underlying base information from areal imagery and/or topography. Thereby, the geospatial mass-market application makes use of distributed Web Services to get the latest information (extracted from sensor data & feature data) for each user request and processes it real-time using OGC WPS. Figure 5 depicts the result of the configured process in uDig and the same process accessed through Google Earth.

7. Conclusion & Outlook

The presented approach enables the seamless integration of sensor information into geospatial mass-market applications by the means of OGC WPS. This is promising, as processes can generate geo-information and especially sensor information, which is required for instance by risk management scenarios. As described in Section 4, the WPS interface specification meets all the requirements for the integration into geospatial mass-market applications. By enabling KML support and the support of the HTTP-GET Execute operation the WPS is capable to be integrated in any geospatial mass-market application. Moreover, as WPS provides the means to extract sensor information from sensor data and to provide this information in KML format, it allows the user to have access to the Sensor Web as such.

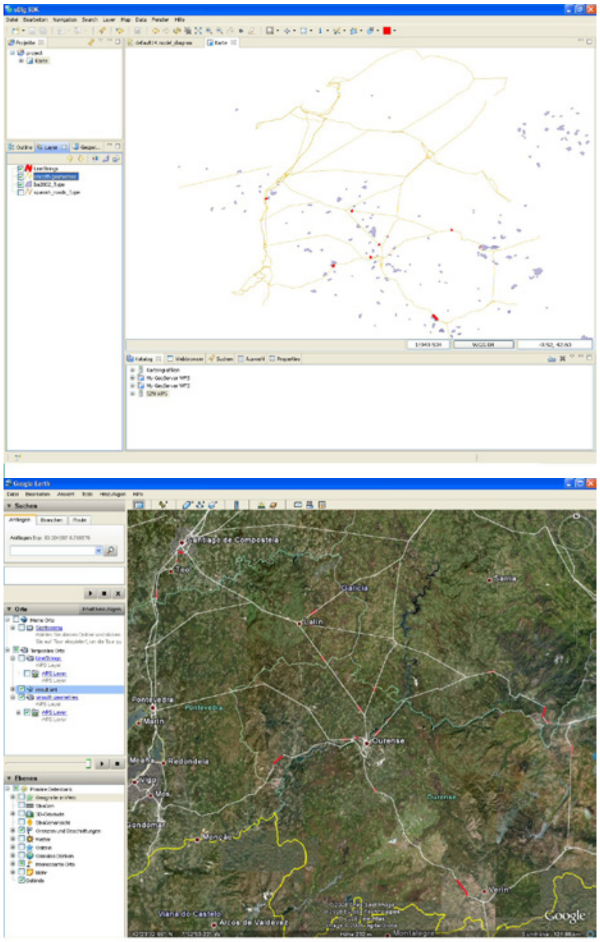


Figure 5: Screenshots of the configured processes in uDig (top) and exported to Google Earth (bottom) – simplified roads & affected road infrastructure (red).

The presented approach is two-fold, as it first allows an expert user to configure the process within an expert client environment (such as uDig). At second, the exported process can be consumed by a geospatial mass-market application such as Google Earth.

The applied use case scenario demonstrates the necessity and applicability of the developed approach for a risk management scenario. Without the integration of sensor information into such applications, citizens would be unaware about current information and could not act accordingly in times of danger. The visualization of sensor information, such as affected road infrastructure combined with static satellite imagery or topography, provides sufficient information to the ordinary user regarding the aimed scenario. Additionally, the approach is interesting for research communities, which need to exchange latest research results in terms of process results (e.g. latest results of climate change models).

The approach is scalable as the sensor information can be integrated as dynamic or static processes (Section 4). It is important to note that the presented approach is fully compliant with the applied standards (KML, WPS, SOS & O&M), without amending or modifying any interfaces or encodings. Overall, the specifications for the OGC WPS interface and family of specifications in the context of the Sensor Web as well as the dynamic KML have shown great flexibility to enrich information on the currently evolving GeoWeb by enabling to integrate sensor information into geospatial mass-market applications.

In particular, the *NetworkLink* feature of KML and the capability of WPS to process referenced data (i.e. data sent by reference), allows the geospatial mass-market applications to integrate service chains accessing the latest data. This is a key aspect in developing architectures for risk management scenarios in the future.

As explained by [33], the performance of web-based geoprocesses as integrated into geospatial mass-market applications is of major interest in the future. Adequate performance is crucial for the usability of the application and opens a new market beyond enterprise applications for the Geospatial Web. Also, the integration of more complex and intelligent geoprocess chains is subject to research.

References

[1] OGC, *OpenGIS Web Processing Service*, Open Geospatial Consortium, 2007.

[2] T. Foerster, B. Schaeffer, J. Brauner, and S. Jirka, "Integrating OGC Web Processing Services into Geospatial Mass-Market Applications," *International Conference on Advanced Geographic Information Systems & Web Services, 2009.*, Cancun, Mexiko: IEEE, 2009, pp. 98-103.

[3] P. McFedries, "The new geographers," *IEEE Spectrum*, vol. December 2007, 2007, p. 96.

[4] A. Turner, *Introduction to Neogeography*, O'Reilly, 2006.

[5] M. Goodchild, "Citizens as Voluntary Sensors: Spatial Data Infrastructure in the World of Web 2.0," *International Journal of Spatial Data Infrastructures Research*, vol. 2, 2007, pp. 24-32.

[6] OGC, *OGC KML*, Open Geospatial Consortium, 2008.

[7] T.M. Smith, O.K. Norman, and V. Lakshmanan, "Utilizing Google Earth as a GIS platform for weather applications," *22nd International Conference on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology*, Atlanta, US: 2006.

[8] C. Kiehle, "Business logic for geoprocessing of distributed data," *Computers & Geosciences*, vol. 32, 2006, pp. 1746-1757.

[9] B. Stollberg and A. Zipf, "OGC Web Processing Service Interface for Web Service Orchestration - Aggregating Geo-processing Services in a Bomb Threat Scenario," *Proceedings of Web and Wireless Geographical Information Systems*, Heidelberg: Springer-Verlag, 2007, pp. 239-251.

[10] T. Foerster and J.E. Stoter, "Establishing an OGC Web Processing Service for generalization processes," *ICA workshop on Generalization and Multiple Representation*, Portland, Oregon, USA: 2006.

[11] A. Friis-Christensen, N. Ostlander, M. Lutz, and L. Bernard, "Designing Service Architectures for Distributed Geoprocessing: Challenges and Future Directions," *Transactions in GIS*, vol. 11, Dec. 2007, pp. 799-818.

[12] D. Roman and E. Klien, "SWING - A Semantic Framework for Geospatial ServicesRoman," *The Geospatial Web*, A. Scharl and K. Tochtermann, eds., London, UK: Springer, 2007, pp. 229-234.

[13] M. Botts, G. Percivall, C. Reed, and J. Davidson, *OGC Sensor Web Enablement: Overview And High Level Architecture*, Wayland, MA, USA: OGC, 2007.

[14] M. Botts and A. Robin, *OpenGIS Sensor Model Language (SensorML) Implementation Specification 1.0.0*, Wayland, MA, USA: OGC, 2007.

[15] S. Cox, *Observations and Measurements - Part 1 - Observation schema 1.0.*, Wayland, MA, USA: OGC, 2007.

[16] Cox, *Observations and Measurements - Part 2 - Sampling Features 1.0*, Wayland, MA, USA: OGC, 2008.

[17] S. Havens, *OpenGIS Transducer Markup Language (TML) Implementation Specification 1.0.0*, Wayland, MA, USA: OGC, 2007.

[18] A. Na and M. Priest, *Sensor Observation Service 1.0*, Wayland, MA, USA: OGC, 2006.

[19] I. Simonis, *OGC Sensor Alert Service Candidate Implementation Specification 0.9*, Wayland, MA, USA: OGC, 2006.

[20] I. Simonis and P.C. Dibner, *OpenGIS Sensor Planning Service Implementation Specification 1.0*, Wayland, MA, USA: OGC, 2007.

[21] I. Simonis and J. Echterhoff, *Draft OpenGIS Web Notification Service Implementation Specification 0.0.9*, Wayland, MA, USA: OGC, 2006.

[22] S. Falke, E. Dorner, B. Dunn, and D. Sullivan, “Processing Services in Earth Observation Sensor Web Information Architectures,” *Proceedings of the Earth Science Technology Conference 2008*, College Park, Maryland: NASA, 2008.

[23] S. Sivasubramanian, G. Pierre, M.V. Steen, and G. Alonso, “Analysis of Caching and Replication Strategies for Web Applications,” *IEEE Internet Computing*, vol. 11, 2007, pp. 60-66.

[24] T. Foerster and B. Schaeffer, “A client for distributed geo-processing on the web,” *W2GIS*, G. Tayler and M. Ware, eds., Springer, 2007, pp. 252-263.

[25] B. Schaeffer and T. Foerster, “A Client for Distributed Geo-Processing and Workflow Design,” *Journal for Location Based Services*, vol. 2, Sep. 2008, pp. 194-210.

[26] V. Olaya, “SEXTANTE - the free geoprocessing library,” Notti: University of Nottingham, 2009.

[27] J. Brauner and B. Schaeffer, “Integration of GRASS functionality in web based SDI service chains,” Cape Town, South Africa: 2008, pp. 420-429.

[28] B. Schaeffer, “Towards a Transactional Web Processing Service (WPS-T),” *Proceedings of the 6th Geographic Information Days*, E. Pebesma, M. Bishr, and T. Bartoschek, eds., Muenster, Germany: Institute for Geoinformatics, 2008, pp. 91-116.

[29] B. Baranski, “Grid Computing Enabled Web Processing Service,” *Proceedings of the 6th Geographic Information Days*, E. Pebesma, M. Bishr, and T. Bartoschek, eds., Muenster, Germany: Institute for Geoinformatics, 2008, pp. 243-256.

[30] A. Friis-Christensen, L. Bernard, I. Kanellopoulos, J. Nogueras-Iso, S. Peedell, S. Schade, and C. Thorne, “Building Service Oriented Application on top of a Spatial Data Infrastructure - A Forest Fire Assessment Example,” *Proceedings of the 9th Agile Conference on Geographic Information Science*, Visegrad, Hungary: 2006, pp. 119-127.

[31] S. Yassemi and S. Dragicevic, “Web Cellular Automata: A Forest Fire Modeling Approach and Prototype Tool,” *Cartography and Geographic Information Science*, vol. 35, 2008, pp. 102-115.

[32] J. Black, C. Arrowsmith, M. Black, and W. Cartwright, “Comparison of Techniques for Visualising Fire Behaviour,” *Trans*, vol. 11, 2007, pp. 621-635.

[33] J. Brauner, T. Foerster, B. Schaeffer, and B. Baranski, “Towards a Research Agenda for Geoprocessing Services,” *12th AGILE International Conference on Geographic Information Science*, J. Haunert, B. Kieler, and J. Milde, eds., Hanover, Germany: IKG, Leibniz University of Hanover, 2009.

Multimodal Robot/Human Interaction for Assisted Living

Ray Jarvis
Intelligent Robotics
Research Centre
Monash University
Wellington Road
Clayton Vic 3168
+ 61 3 9905 3470

Ray.Jarvis@eng.
monash.edu.au

Om Gupta
Intelligent Robotics
Research Centre
Monash University
Wellington Road
Clayton Vic 3168
+ 61 3 9905 3410

Om.Gupta@eng.
monash.edu.au

Sutono Effendi
Intelligent Robotics
Research Centre
Monash University
Wellington Road
Clayton Vic 3168
+ 61 3 9905 3410

Sutono.Effendi@eng.
monash.edu.au

Zhi Li
Intelligent Robotics
Research Centre
Monash University
Wellington Road
Clayton Vic 3168
+ 61 3 9905 3410

Zh.Li@eng.
monash.edu.au

Abstract

This paper outlines the framework of a complex system to demonstrate multimodal spatial and transactional intelligence in a robot which autonomously supports aged, frail, or otherwise disabled people in a domestic assistive technology context. The intention is that the robot be able to navigate around a known multi-room environment along optimal, collision-free paths in search and retrieval of requested objects such as spectacles, books etc. and must also be capable of tracking and following humans and of reminding them of times for meals, medication etc. and to lead disoriented subjects to their meal place at appropriate times and even dispense medication, if necessary. The modes of communication interchanges with the supported human include spoken speech and gestures (including eye gaze direction) within the context of situational analysis which accommodates recent history, temporal factors and individual user behavioural models. This paper provides an overview of an ambitious research project in its early stages, describes many components developed to date and outlines future work.

Keywords

Intelligent robotics, assistive technology, scene analysis, robot navigation, gesture recognition, human/machine interaction.

1. Introduction

As robots emerge from the structured industrial environments they have habituated for some time into the relatively unstructured spaces of the built and natural world it is clear that they require increasing levels of intelligence, informed by rich sensory sources, to survive, adapt and serve humans, where humans and robots mix freely. In assistive technology environments, where the served humans are perhaps aged, infirm or otherwise disabled, either mentally or physically, the useful interactions between robots and humans need to be particularly sensitive and sophisticated, since no expert knowledge of the technology can be assumed to reside with the served humans. Also, in some cases, even the basis of normal human to human interaction may be partially undermined by physical or mental dysfunction.

Two quite distinct, but functionally linked, types of intelligence need to be mastered by the robot. The first is 'spatial intelligence', which is the understanding of how the working environment is structured in terms of geometry, occupancy and functional designations (eg. kitchen, bedroom etc.) and how to deal with (find, recognise, handle) common objects in it (eg. cups, knives, forks, books, spectacles etc.) in fulfilment of useful duties. This type of intelligence includes the capacity to achieve goal directed path planning and following, obstacle avoidance, collision-free robot arm trajectory planning and object recognition/manipulation. Each of these requirements are serious challenges in themselves. Having them all function together in a smooth operation to achieve some overall mission is at least one order of magnitude higher in complexity, in operational management, context integration, conflict resolution and hierarchical planning terms.

Spatial intelligence, as defined above, has been well researched, in its various components, by the robotics research community over the last two decades or so (Jarvis and Byrne, 1987; Jarvis, 1997; Durrant-Whyte and Guivant, 2000; Jafari and Jarvis, 2005 ; Rawlinson and Jarvis ,2007), with various levels of maturity and reliability being achieved in each.

The second type of intelligence, which must also be mastered to achieve the overall goal of serving humans correctly, safely, and with grace, is 'transactional intelligence'. By 'transactional intelligence' is meant the understanding of how to communicate with humans to be able to correctly interpret and carry out their wishes and to clarify uncertain or detailed aspects of the task at hand. The interaction is a kind of negotiation process which should result in purposeful and correct interpretation and action, with intermediate interchanges sometimes needed to resolve ambiguity whenever it arises. Understanding the intent of the human and how to go about fulfilling it are the essential requirements within the assistive technology context. Accommodating the practical limitations imposed by the physical state of the environment (and the objects within it) and the capabilities of the robot is crucial to these ends. Understanding what a human intends is often fraught with considerable ambiguity. Resolving this ambiguity within the above constraints is the basic goal of 'transactional intelligence' in this context.

Multimodality provides the crucial key, which unlocks how best to resolve ambiguities of interpretation in both the spatial and transactional components of this project. Multiple sensor modes, including stereoscopic and panoramic vision, tactile sensing, laser range finding and range cameras are deployed to resolve issues concerning how to navigate efficiently through unoccupied space (avoiding fixed and dynamic obstacles), searching for and recognising target objects, manipulating objects (and carrying them), and finally putting them before or in the hands of the requester. Likewise, multiple communication modes, including spoken language understanding, gesture recognition, face recognition and gaze direction analysis, are used to achieve 'transactional intelligence' in the context of environmental constraints (e.g. where the robot can go, where objects are, what things can be picked up etc.), individual behavioural modes of the user (once recognised) and maybe even the time of the day (e.g. the approach of meal or medication times).

The following section touches briefly on related work. The next outlines the robot's attributes, its control structure, its sensory capability, and the methods of achieving localisation (i.e. position/pose), environmental modelling, path planning, obstacle avoidance, object recognition, and manipulation. This is followed by a section on the way in which various modes of human/machine communication are to be combined to resolve ambiguity, including the possibility of querying the human to help resolve and/or refine intention. Then follows a section describing a functional subdivision of project tasks to enable various aspects of practical implementation to be put into operation. The next section describes progress so far and the type of experiments being planned for the future. References to papers developed in the author's laboratory have been deliberately given emphasis to indicate the background work supporting this current project. This paper arose out of a previous conference presentation (Jarvis, 2009).

2. Related Work

An increasing amount of research work has recently been focussed on the area of assistive technology with the realisation of an anticipated large number of old/frail people likely to be inhabiting most Western nations over the next twenty years and the corresponding reduction of younger/fit people available to look after them. Many proposed solutions look to modern technology for support. The overall scope of assistive technology is very wide. The more traditional work has been to provide smarter wheelchairs which can provide sensor informed assistance to their users, allowing them to be able to move about purposely with both independence and safety (Jarvis, 2002; Hu et. al., 2007) and to use robotics to provide physical prosthesis (Carrozza et. al.,2001; Pons et. al.,2004)and therapeutic manipulation(Volpe et. al.,2009). More recently there has been a keen interest in tracking (Chakravarty and Jarvis, 2006) old/sick/feeble people to check whether their movement habits are changing, possibly as a harbinger of some serious physical or mental deterioration or to indicate when a fall has occurred (Lee and Mihailidis, 2005) and requires immediate attention. Also, there is an interest in the electronic monitoring of blood pressure, blood sugar and heart rhythms (Gao et. al. 2005) which can be relayed directly to a medical centre for attention if required. Some of this effort has strong associations with security based surveillance (Kanade et. al., 1997), particularly if purely passive methods of observation are used. Combining robotic wheelchair navigation with on-board robot arm manipulation (Prior, 1990) has also attracted research interest; the requirement for robotic hand/eye coordination (Hagar and Chang, 1995) and pattern recognition is clearly evidenced here. In the field of robotic companions (Dautenhahn et. al. 2006), there has also been implications regarding their application for assistive technology as well as entertainment (Tamura et. al. 2004). More generally, there has been considerable work on improving human/machine interaction (BuŸ́hler et. al., 2002) with a focus on more natural modes of indicating human needs and intentions; clearly, whilst these efforts can enhance the way the general population communicate with computers, they have particular importance in assistive technology applications, especially where the user has limited mental capacities.

In our project, to be outlined in what follow, we have combined the areas of multimodal human/machine interaction with both robotic hand/eye coordination and robot navigation. This holistic approach is fairly unique, as represented in the literature. Whilst mobile robot navigation and robotic hand/eye coordination have long been central to Intelligent Robotics research, until recently, questions relating to human/machine communications have mostly been of interest to the Human-Machine Interaction (HMI) research community. Now that robotics researchers have realised the importance of this topic, they have been enthusiastic in their inclusion of these human-centric elements in their research. Nevertheless, the combination of HMI, navigation and robotic hand/eye coordination in the one project is rare but is the main emphasis of our project which sets it apart from other research efforts.

3. The Robot, Sensors, and 'Spatial Intelligence' Methodologies

The robot [See Figures 1 (a) and (b)] consists of two main parts. The mobile base is simply an adapted electric wheelchair motor/gear/control set, which can carry a human payload for up to six hours between battery charging. It is differentially steered with castor wheels at front and back (for stability) and can turn on the spot. A UMI six degree of freedom robot arm is mounted on the mobile base. This arm is safe to use in the vicinity of humans since it is slow, relatively weak and has plastic coverings. It has an extensive vertical movement axis, which makes it ideal for retrieving objects off various height tables and shelves and has a simple two fingered gripper. The control schematic is shown in Figure 2. An onboard laptop computer drives a serial four port server. One port sends commands to the robot manipulator and a second drives a 32 channel servo motor (hobby type) controller.

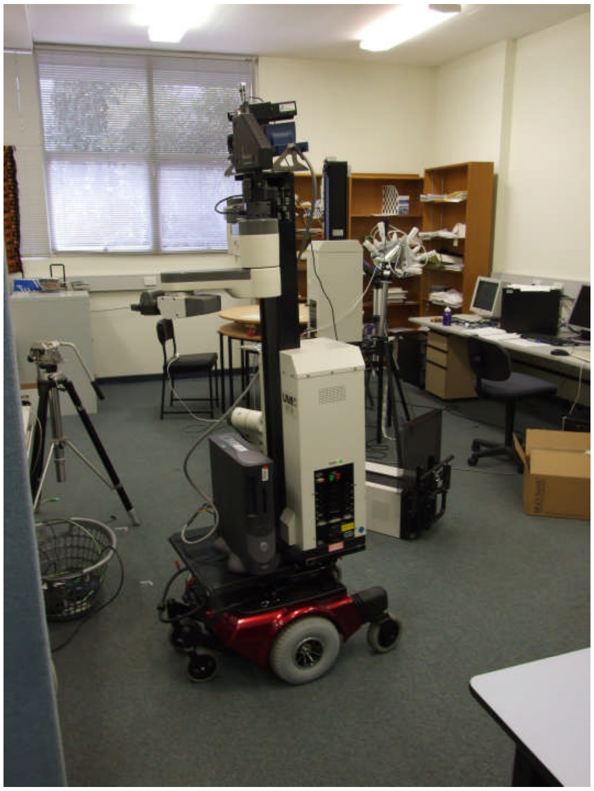


Figure 1(a) Instrumented Robot with Manipulator

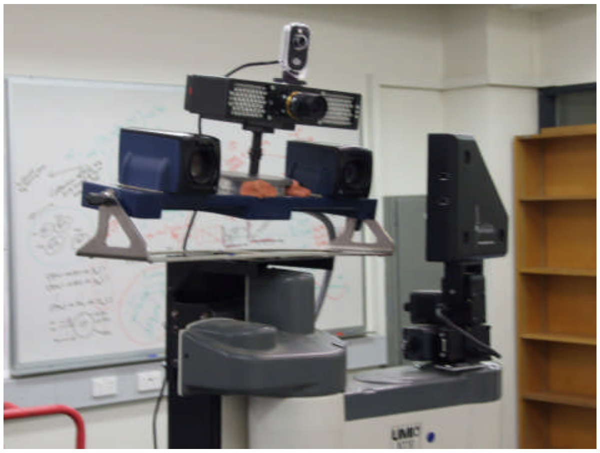


Figure 1(b). Gaze Tracker, Stereo and Range Cameras

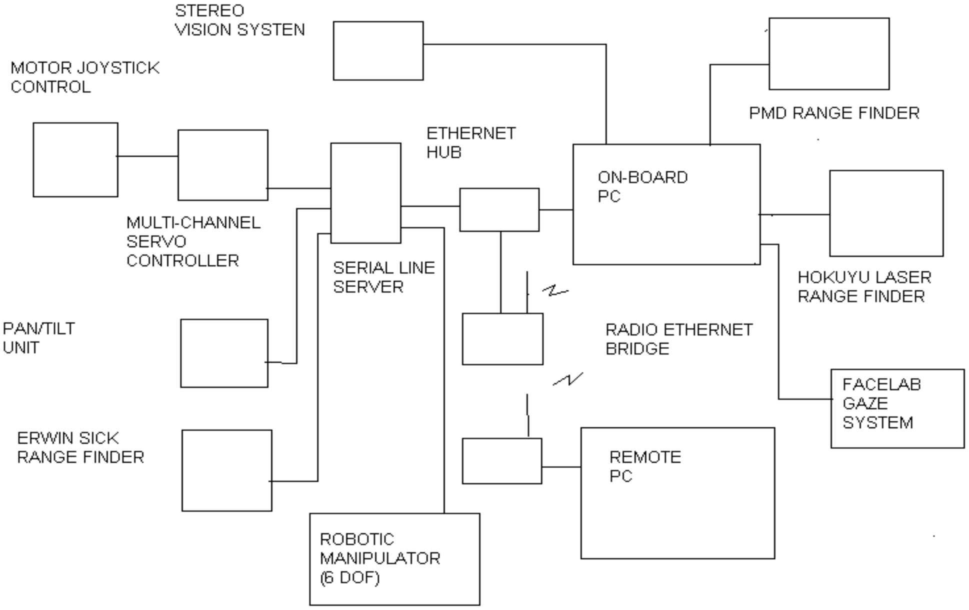


Figure 2. Control/Sensor Data Acquisition Schematic

Two of these channels control the joystick normally used to drive the wheelchair base. A third serial line driver port can control a pan/tilt head for directing a stereo camera/colour camera system towards various targets and the fourth collects range data. Sensors onboard the robot include a Hokuyu line scan laser scanner to be mounted on the robot manipulator hand and an Erwin Sick laser range finder low at the front of the robot, a colour panoramic camera at the very top and a stereo gaze direction analyser (SeeingMachine's Facelab), currently pictured at the base of the robot but to be relocated at head height. A simple localisation scheme will use panoramic vision mapping with pre-scanned laser range/finder camera maps of the working environment acquired by a Riegl LMS Z420i scanner/imager [See Figure 3.].

Figure 3(a). Riegl Laser Range Scanner/Camera

The detailed 3D colour map thus acquired will be hand annotated to indicate functional spaces (eg. kitchen, bathroom etc.) and functional large objects (tables, shelves, refrigerator etc.). The overhead camera will be able to note changes of position of chairs and tables for navigation purposes but human intervention will be required for changing functional annotations if such is required over time. Distance Transform (Jarvis, 1985; Jarvis, 1994) path-planning methodology is to be used for global planning with semi/reactive obstacle avoidance adapted from an earlier project (Jarvis, 2000). Further details follow.

Scene analysis of objects on table tops (to determine the existence, location, pose and identity of relevant objects) will be carried out using a combination of laser range finding (Hokuyo laser scanner) and passive stereo vision (Pointgrey's Triclops/Bumblebee). Details follow. Some limited tactile sensing between the robot manipulator's grippers is envisaged to confirm the identity and actual grip of objects to be manipulated and carried. A typical intention such as 'please find my blue mug, which is usually in the kitchen and brings it to me' would be supported by this system. The location of humans would also be tracked so that the robot can approach them and or follow them if they are moving, as the situation may require (Chakravarty and Jarvis, 2006). For example, it makes sense to bring a requested item to the requester, even if he/she has moved since the request.

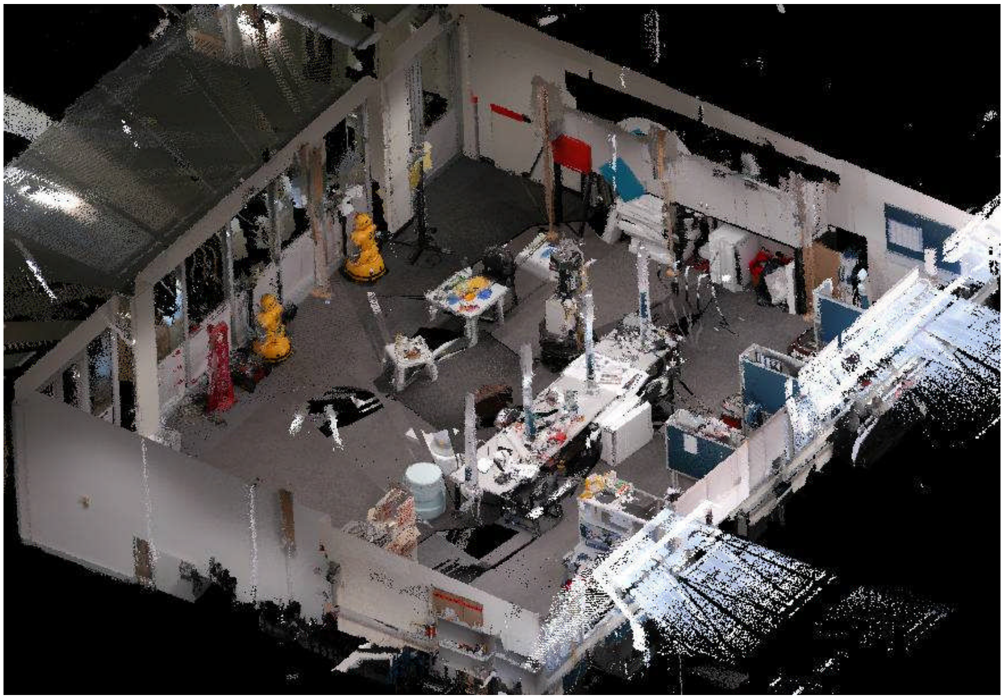


Figure 3(b). Range/Colour Scan of Laboratory

4. Multimodal Human/Robot Transactions

The dominant modes of human/robot communication for this project are spoken language understanding and gesture recognition (including eye gaze). A complex aspect of the language understanding component involves the use of dialogue history and user models as disambiguating knowledge sources and is thus dynamic (anytime) rather than static process. A simple 'first try' version of this combination (speech/gesture) has been published (Harte and Jarvis, 2007). Also included are to be face recognition to establish the identity of the user so that her/his particular habits of communication may be accommodated, and to establish authorisation and 'attachment' for a period

sufficient to complete a simple task following instruction. Face recognition can also be used to check whether a visitor is a familiar one or a stranger who perhaps should not be given free entry without further checks by authorised personnel. A gaze direction system (Facelab), previously used to help a disabled user to navigate a wheelchair (Jarvis, 2002), is also to be used to refine gesture interpretation which will mainly be concentrated on arm and hand movement (e.g. say when looking at an object being roughly pointed at). Details follow. The overall schema for the project is shown in Figure 4. However, it is hard to use such a conceptual breakdown as an implementation guide. A simpler way of resolving ambiguities of human intention than is shown in Figure 4. will be used in the first instance. If several modalities (say speech and gesture) are in conflict as to the user’s intention, the confidence weighted probabilities of interpretations for each mode separately will be used as votes, looking for feasible alternatives and the most likely correct one. If there remains high uncertainty after this process, the user can be asked for clarification or to simply repeat the request more carefully and perhaps slowly. Clearly we would like to avoid these kind of clarification requests as they would annoy the users. Hopefully, if we are able to associate certain communication habits for individuals, we can minimise their use.

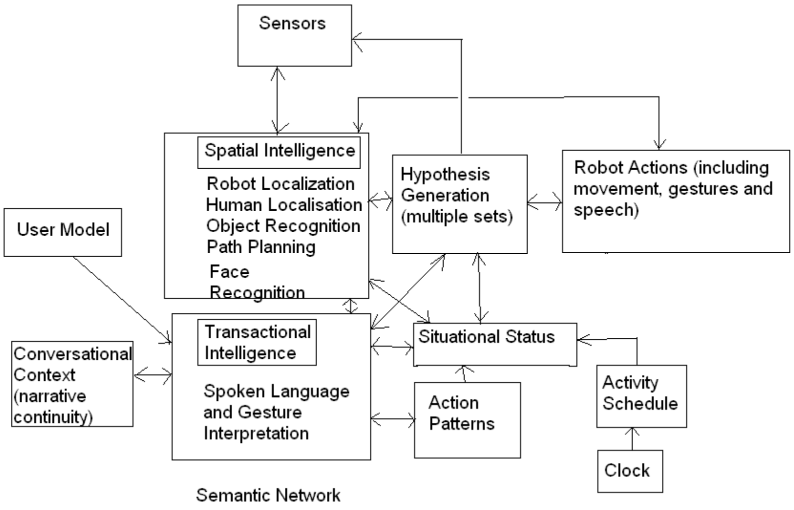


Figure 4. Semantic Network Schematic

5. Task Subdivisions for Project Development

In order to allow two fairly distinct groups of researchers, one predominantly working in robotics, the other on language understanding (within the context of the robot's capability and pertinent human requests for assistance), a functional subdivision of the system has been developed with the intention of being able to integrate the two teams' efforts in a seamless way. Three interlinking nodes can be defined in this task subdivision plan. Firstly, one node will be responsible for the generation of hypotheses of the intent of the user and possible reasonable robot responses and tasks, and the resolution of ambiguity towards discovering a dominant hypothesis. This system will use current dialog, dialog history and user modelling as well the spatial (existence and probable location of objects) and temporal contexts(time of day and scheduled events), together with lists acceptable task possibilities. Some kind of extended negotiation (transaction) will sometimes be needed to obtain some convergence between what the user wants and what is reasonable for the robot to do. Some clarification questions may also have to be posed for this purpose and feasible alternatives may be offered for selection and acceptance.

The second node embodies the capabilities of the robot, its various skills, such as collision-free navigation in dynamic environments (details follow), scene analysis, hand/eye coordination and the means of triggering tasks relevant to the assistive technology domain. Inputs instructions to this node from the hypothesis generation/resolution and task definition node are to be unambiguous and reasonable (but may prove to be unfeasible in a particular instance) and ready for immediate execution. The robot will then attempt the assigned task and report success or failure of completion, with the possibility of specifying one of severable failure modes (such as navigation passage blocked, object can not be found , object found but inaccessible, batteries exhausted etc.).

The third and most vital node concerns the database of spatial geometry (room dimensions, locations, fixed furnishings etc.), object lists (specific as well as generic), estimations of probable location, time stamps of all actions taken which modify the database and the source of the modification (robot/sensor system or hypothesis generator/ambiguity resolver system). Flagging all modifications for validation would be a useful mechanism to support database integrity.

The initial database would be based on dense 3D geometric and surface colour data from the Riegl laser range/colour image scanner which would collect information off-line as a habitat specification , done only once beforehand. This raw scanner data will be hand annotated to label all relevant spaces, furniture, fittings, objects and utilities (stoves, fridges, heaters, toasters, kettles etc.) and extract size, colour and location data for the database. Fixed and movable items will be so classified as would be specific (e.g. a particular book) and generic items (e.g. regular mugs, plates etc.). As object moving actions and/or sensor observations dictate, the database would be modified to reflect the new reality. Clearly all proposed changes will need validation before execution. Uncertainties can be specified probabilistically. Whilst one could consider the robot (plus sensors) being able to construct this database piece by piece as it moves about, the idea of using pre-scanned data is much more practical and almost certainly more accurate. It also permits advancing the project to the transactional intelligence development stages with minimal delay.

We intend to move towards an integrated system where the time of day, the likelihoods of various objects being at various locations (normal expectations plus history of use), the behavioural particulars of a user, the history of language dialogues and gestures, the risk factors associated with making incorrect interpretations and the nuisance value of too many clarification queries can all be taken into account within a working physical system where a real robot carries out useful tasks for an aged, fragile or otherwise impaired human in the familiar surroundings of a home-like environment, where people and robots freely and safely mix.

6. Progress to Date and Plans for the Future

The entire project is a quite ambitious and complex one with many interlinked components. However, many of these have already been addressed in earlier projects and are readily adapted to this one. The Distance Transform methodology (Jarvis, 1985; Jarvis, 1994) for global path planning has been fully investigated and applied successfully in conjunction with barcode scanned localisation on an indoor robot capable of finding its own collision-free way around an obstacle strewn environment (Jarvis, 1997). Localisation using a panoramic vision system and image matching against pre-scanned 3D plus colour detailed maps of the environment has also been demonstrated (Jarvis, Ho and Byrne, 2007), yet the system of overhead panoramic camera localisation is preferred for simplicity. Fusing simple speech recognition with primative gesture and object recognition has also been demonstrated (Harte and Jarvis, 2007). Gaze tracking and dynamic obstacle reactive avoidance in relation to a semi-autonomous wheelchair project has also been employed successfully (Jarvis, 2002) as has human target tracking (Chakravarty and Jarvis, 2006) and face recognition (Axnick and Jarvis, 2005). The challenge is to combine all these previously tested systems into a common framework and to provide semantically related clues and sophisticated ambiguity resolving methodology to meet the requirements of the assistive technology environment targeted.

It is intended (an already commenced PhD. Research project) that a sophisticated gesture capture and recognition system be developed for markerless subjects, using a combination of colour video and range camera sensors (details follow), fusing these two sources of spatial data to extract the dynamic parameters of link movements on a skeletal model (upper torso and head only, initially) of a human and then to train this system to bind sequences of movement to intended communication tokens (for each individual subject). Once an individual is identified using face recognition their raw gesture sequences can be interpreted in a customised way, since different people often have differing ways of specifying intention which may be sometimes culturally dependent as well as individual. Gaze direction vectors and mouth movement detection (for verifying that the person fronting the robot is speaking) will also be extracted.

In what follow, some details of progress to date are reported as extensions of general approach material presented earlier in the paper. These include navigational, scene analysis and gesture recognition work completed so far.

A. Mobile Robot Navigation

The three essential sub-system requirements for autonomous mobile robot navigation are localisation (determining the location and orientation of the robot), environmental modelling (capturing relevant details of the working environment) and path planning (determining the collision-free movements of the robot through the environment from a start point to a nominated goal).

Localisation can be performed using simple odometry (from measuring wheel rotation) with a known starting position/orientation but, due to slippage, imperfect circularity of wheels, undulation of the floor and wheel shape variations under load, errors accumulate incrementally and can, eventually, render the evaluated location/orientation unusable. Beacons at known locations can also be used but this requires careful, possibly tedious, site preparation. Natural landmarks are an alternative, but the computational load is quite high, depending on what on-board sensors are used. In our project we have the advantage of a pre-scanned environment (using a Riegl LMS Z420i laser range scanner/camera) within which particular objects (eg. doors, tables, fridge, book shelves, cupboards etc.) can be annotated and localisation can be determined using an onboard panoramic vision system. However, we have chosen to use a fixed high vantage point panoramic video camera that can recognise the position and orientation of the robot as well as track people and note variations in the obstacle strewn space, both slow and fast changing. Thus the path planning can take into account both static and dynamic obstacles and also accommodate the positions and movements of people, perhaps even using predictions of human movement intention into consideration.

Environmental modelling requires either prior knowledge of the working environment (maps, plans, scans etc.) or a means of incrementally constructing a model using on-board sensors whilst the robot moves around the environment. A considerable body of published work addresses the notion of combining localisation with environmental mapping (Simultaneous Localisation and Mapping – SLAM) (Durrant-Whyte and Guivant, 2000) but, until the map is sufficiently complete, optimal path planning cannot be carried out. In our situation, a complete detailed scan of the environment is taken just once using a Riegl LMS Z420i laser range scanner/camera. The particular advantage of this approach (in contrast to SLAM) is that all functional details of the environment (tables, doors, stoves, cupboards) can be annotated in the data for proving goals for subsequent path planning in accordance to the task required to be carried out by the robot. The overhead panoramic camera mentioned in the previous paragraph supplements this data to include updates and note dynamic aspects, particularly the movement of people.

Path planning has aims, firstly, to arrive at a nominated location without obstacle collision and, secondly to do so efficiently as determined by some optimality criterion. Both static and dynamic obstacles should be avoided, perhaps some busy traffic areas a voided if possible as a courtesy to humans and perhaps even the approach to a human made unobtrusively yet without startling the human from behind included in the path planning strategy. From amongst the large number of path planning strategies available we have chosen the Distance Transform approach since all the aspects mentioned above, such as static and dynamic obstacle avoidance preferred no-go zones human movement predictions, human approach preferences etc., can be easily taken into account and the path plan re calculated frequency when required.

The Distance Transform (DT) path planning algorithm is very simple to construct. Details are provided elsewhere (Jarvis, 1994), but the gist of approach can be easily described:

1. Construct a tessellated floor map with each cell representing a 'floor tile' of appropriate dimensions relative to the robot dimensions (say, a 2x2 set of tiles approximately the 2D size of the robot). Each cell should contain a positive cost representing the cost of entering that cell, obstacles being given an infinite cost. Preferred no-go zones can have high costs. It is even possible to have the cost of entering a cell depend on which neighbour the entrance comes from, thus allowing preferred directions and one way only constructs but this will not be included here.

2. In a separate map same structure as the cost map, described above, goal point cell is set to zero and all free space (space not occupied by obstacles) set to a large number). This is called the DT map.

3. In a forward raster order (left to right, top to bottom), skipping over obstacle cells, replace the constants of each cell by the least of recently visited neighbour's (three above and one to the left) number plus the cost (from the cost map) of entering that all.

4. In a reverse raster order (right to left, bottom to top) repeat the strategy of 3, noting that recently visited neighbours consist of three below and one to the right.

5. Repeat 3 and 4, above, until no change occurs. Now the DT map is the cost weighted Distance Transform.

6. From any point in free space, the steepest descent trajectory in the DT map leads optimally to the goal. The DT map, itself, is starting point independent.

The starting point independence of the DT map has the advantage that, should the robot wander off or deliberately move off the planned path (to avoid a fast mobbing obstacle not yet in the cost map), it can recover by continuing a steepest descent path from its new location. Since the DT provides the distance to the nearest goal for each free cell, the steepest descent trajectory can be followed from any point in fee space to reach the goal in an optimal path.

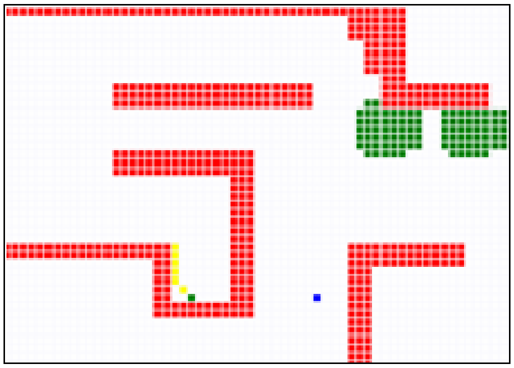
Multiple goals (any one of which when achieved is sufficient) can be included without altering the algorithm (putting zeros in the DT map), the steepest descent trajectory from any point in free-space leading to the goal with the least cost of achieving.

The DT strategy can be extended to any number of dimensions and can also be adapted to cope with the six degree of freedom (or more) movements of robot manipulations using configuration space.

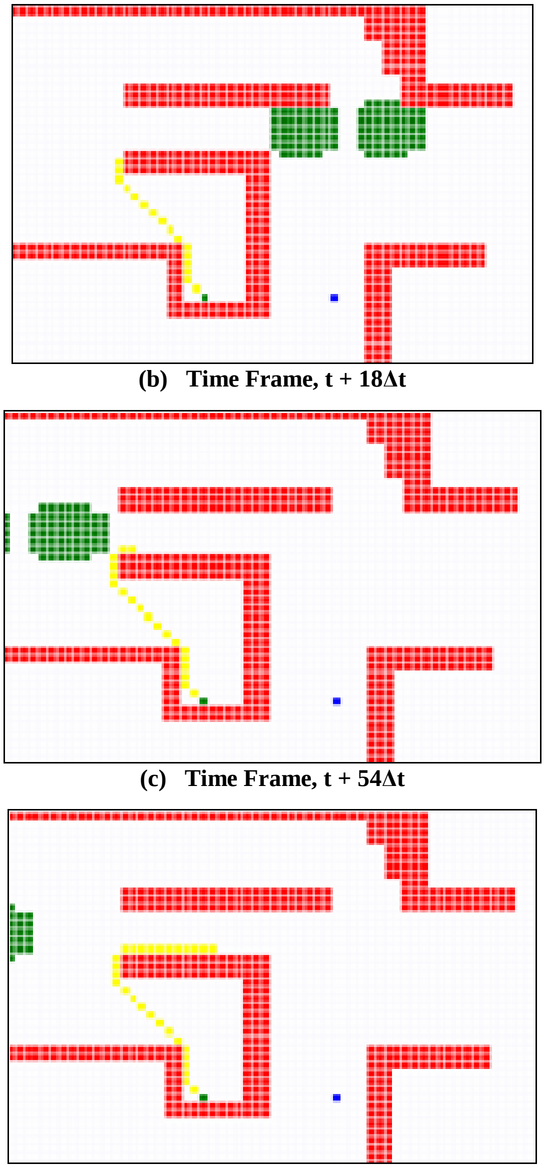
Should one of the dimensions be time a spatio-temporal DT can be generated. The only significant difference when compared to physical dimensions is that, time being irreversible, the construction of the DT only requires one pass on the time dimension.

Suppose a time dimension were added to a two physical dimension tessellated map space. A stack of 2D maps, each representing are point in discreet time, can be piled up, with the top one being at the present and the bottom most one the time count in the future represented the extent of the future factual details of obstacle space or even precautions of such. Cells in the 3D (2D + time) stack marked zero represent goals in time/space (rendezvous). If these are all fixed in position (on a vertical pole through the stack) it means that a nominated position can be reached at any time (of course we still want cost optimality in reaching it). Isolated goals must be reached precisely in the appropriate time interval. A continuous strength of goals through time is like trying to meet a moving goal. Again a corresponding cost may contain non-negative cost values. If only distance and waiting is costed a simple cost structure where waiting costs one unit, front/back and sideways moves costs 2 units and diagonal moves costs 3 units $\left(\frac{3}{2} \approx \sqrt{2}\right)$. Details are given in (Jarvis, 1994). The algorithmic scan moves from the most future layer backwards in time until the top (present) level, replacing costs at each level in parallel (each cells replacement can be calculated independently). Only one pass is necessary. From the present level start point the steepest descent path through time space leads to the least costly achievable goals. The cost replacement rule is to calculate the cost from each neighbour and the cell itself in the next time interval plus the cost of moving into the cell and choosing the least sum. No physical movement costs one unit for waiting cost (like a taxi charge).

Of course, in the simplest case one must have a perfect prediction of the locations of all obstacles in the future until the time interval represented by the bottom map. However, these costs can be probability values calculated from predicted movement observations. As the robot moves physically in time the future horizon can be extended and repopulated with estimated cost with the DT calculated repeated as open as required. Since the space/time DT is a one pass evaluation and essentially a parallelisable algorithm it can be executed quickly. If the obstacle prediction is based on observed movements and simple expectations of straight line trajectories the future costs can be simply determined. Uncertainty can be modelled using Gaussian or other distribution functions and even the distortions of the spread functions caused by not being able to penetrate fixed obstacles can be taken into account. Some example of dynamic obstacle field path planning are shown in Figure 5.



(a) Time Frame, t + 6Δt



(b) Time Frame, t + 18Δt

(c) Time Frame, t + 54Δt

(d) Time Frame, t + 65Δt

Figure 5. A scenario where waiting is preferred by path planning algorithm.

Figure 5 represents a collection of key snapshots from the simulator at different time slices. In the figure, the green shaded region represents the moving obstacle. When the robot encounters the predicted moving obstacles, it waits until the obstacle passes before continuing towards its destination. Although, it may seem that the moving entity is colliding with the red-shaded static obstacles, this is not the case as both the red-shaded static obstacles and the green-shaded moving entities are dilated in order to avoid getting too close to each other and the robot.

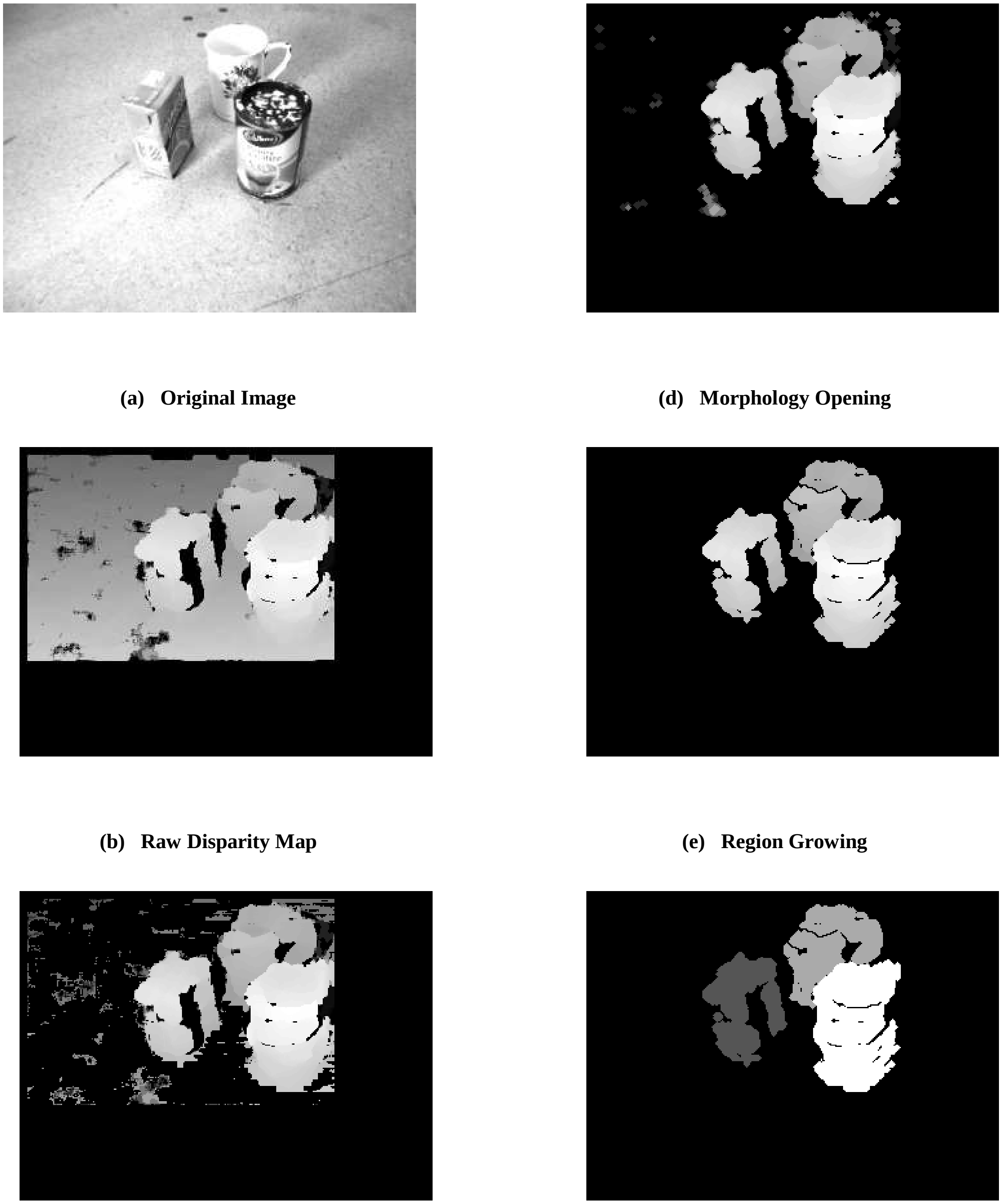
Thus in some scenarios, it can be argued that the optimal route may require a robot to stop and wait approach to avoid moving obstacle.

Unpredictable dynamic changes in the environment can be observed from an overhead panoramic video camera and can be quickly incorporated into the DT planner.

From time to time it may also be required that the annotated map of the environment be adjusted if more permanent changes are to be accommodated. This can be done manually.

B. Robotic Hand/Eye Coordination

When the robot approaches the table or shelf where a target object (eg. cup, fruit, spectacles, pencil etc.) might be found, the local scene needs to be analysed to identify the target object and estimate its position and pose so that it can be grasped and withdrawn from its environment without collision (so that it might be taken to the human requester). Scene analysis concerns describing objects in terms of identity, position/pose and juxtaposition components may have to specify proximity, access, support or containment aspects of the target object in the context of its relevant neighbours. Robotic hand eye coordination's refers to the capability of directing a robot manipulator to manipulated objects in the scene based on it s visual analysis results. In our case, the robot arm is on a mobile robot also carrying the scene analysis sensors. The aim of robot hand/eye coordination in the context of our application is to retrieve the target object without collision. Scene analysis can be particularly difficult if objects in the scene and in a jumble or when some are visually obscuring others. There are two classes of object recognition tasks involved. The first class is where a specific, unique object needs to be found against a database of possibilities. Methodologies based on Scale Invariant Feature Transform (SIFT) (Lowe, 1999) features are ideal for solving this type of pattern recognition problem. A number of localised features are extracted by sight as signatures of the objects they belong to. Finding sufficient matching signatures with respect to a particular object in the database objects (pre calculated) is all that is required to identify a specific object (eg. a particular mug, bottle, pan or spectacles etc.). The object database can have multiple entries of the same object viewed from a variety of positions; this permits recognition even in severely obscured conditions and for varied poses. We use a Triclops (Pointgrey) stereo camera to initially segment the objects in a scene and to identify the supporting plane (eg. table surface). SIFT can then be used to identify the target object if it is present and the stereo disparity (3D data) and segmentation results can be used to construct a virtual bounding box around the identified object to allow grasping to be planned and executed from above (thus reducing the collision problem). A simple example is illustrated in Figure 6.



(a) Original Image

(b) Raw Disparity Map

(c) Ground Plane Removal

(d) Morphology Opening

(e) Region Growing

(f) Object Extraction Result

Figure 6. Stereo Segmentation Analysis Sequence

The second class of recognition problem which needs to be solved in the generic one where the target object is in a class where unique individuals are not represented in the database. Finding an apple or pear or a glass, fork, knife etc. would be examples of the aim for this type of recognition. This is a more difficult problem, which we are still working on, the obscurance problem being a particularly difficult aspect of this task. The simple-minded approach of picking up and separating each object and then attempting to recognise it in isolation is not considered a satisfactory methodology for our application.

C. Gesture Recognition

Gesture recognition is to be one key component of the transactional intelligence aspects of this project. Combining gesture with voice recognition to reduce ambiguity in expressing a task intention to the robot is seen as proving a natural and robust human/machine interface, which could be used by non-technical, possibly frail users. Off-the-shelf face recognition and voice recognition systems are used to support this aim. We adopted a face recognition software package called 'verilook' (version 3.2), which is provided by Neurotechnology. It supports simultaneous multiple face processing in live video and still images. It does not require high resolution images; webcams or other low cost cameras can be used. A quality threshold can be used during face enrolment to ensure that only the best quality face template will be stored into database. VeriLook has certain tolerance to face posture that assures face enrolment convenience: rotation of a head can be up to 10 degrees from frontal in each direction (nodded up/down, rotated left/right, tilted left/right).The voice recognition software we use is called 'Dragon Naturally Speaking' (version 9) provided by Nuance Company. It translates the user's speeches into text. The recognition accuracy can be significantly improved by intensive training. In addition, we can add words or phrases into our own vocabulary set and put emphasis on them in the training stage, so that we can achieve a high recognition rate for the frequently used phrases in our application. By recognising an individual, his/her expected behaviour with respect to voice and gesture communications can be used to simplify and improve the quality of gesture/voice command and dialog recognition.Also, the same person using a fetch command can be the recipient of the fetched object even if they have moved, provided their movements are tracked by the overhead camera. Face recognition can be used to verify that tracking has been carried out correctly. Gestures are useful and a natural way of expressing geometrical and spatial information, particularly pointing and come and go and stop indications.

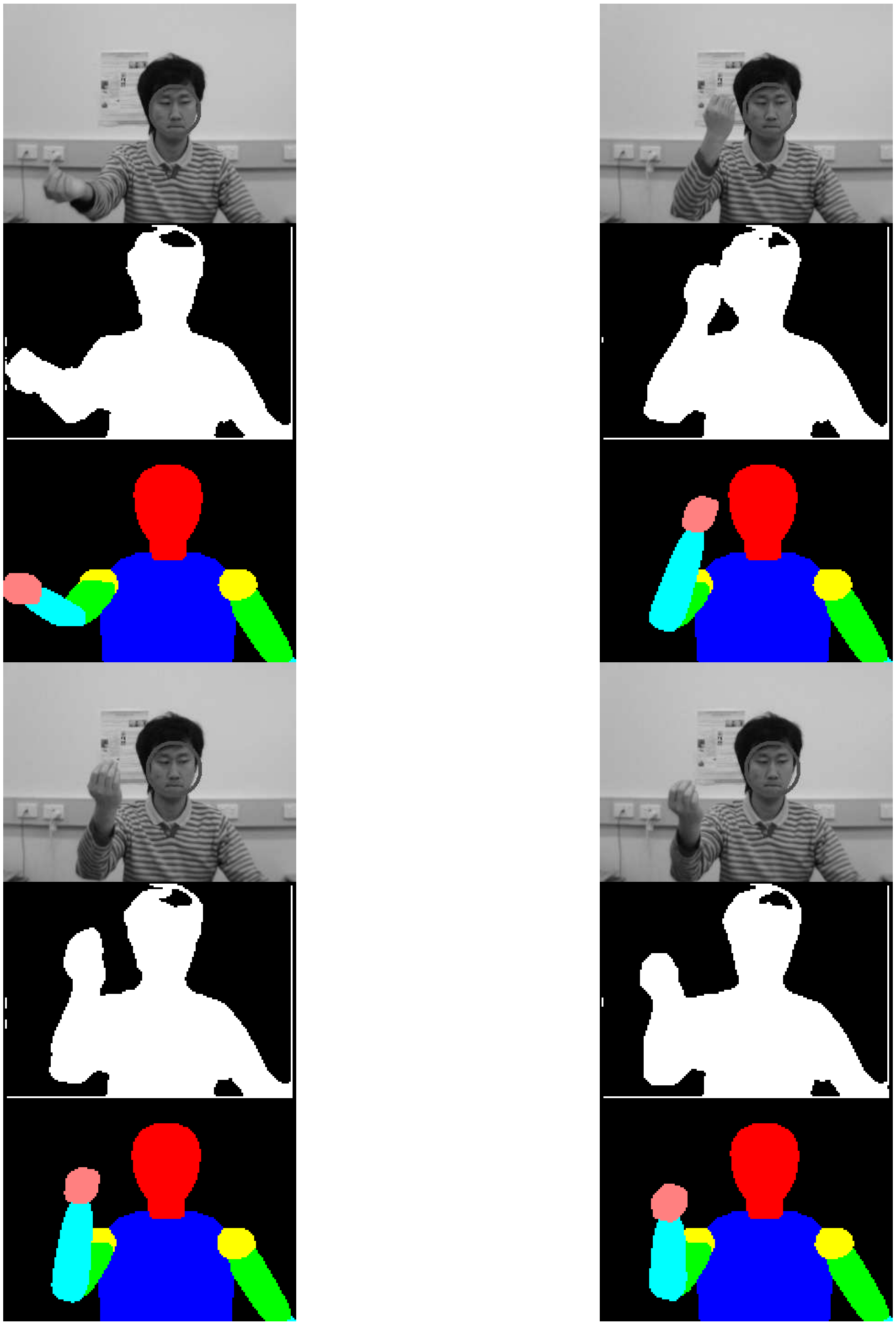
We are using a PMD Technologies range camera which provides frontal surface distances at low resolution (160x120) but high speed. The technology behind this type of camera is infrared laser time-of-flight measurement with each pixel evaluating this measure. A colour web-cam, mounted on the range camera, is also used as it provides higher resolution colour image data. Detecting a face is a first step to determining the position of a person as a preliminary to arm and hand gesture recognition. The resolution of the range camera is too low for reliable face detection so the colour web-cam image is used instead. The algorithm used is based on Hair- like features (Lienhart and Maydt, 2002) and boosted classifiers (Lienhart, Kuranov and Pisarevsky, 2003). Subsequently, the position of the face is found in the range camera data using the SAD (Sum of Absolute Difference) matching method.

We anticipate using a Xuuk's Eyebox 2 camera which detects infrared light reflected through the retinas (exploits the 'red-eye' effect) to count and localise people looking towards the camera to instantly position the vision/range system mounted on the mobile robot in an appropriate position for face and subsequent gesture recognition. We also hope to identify the person who is giving commands to the robot, amongst those facing the robot, by lip movement and audio cues.

When the user's face has been located in the range camera data, the distance between that user and the robot can be estimated. On the assumption that the human body (upper torso, arms and head) can be enclosed in a cubic volume determined by the 3D position, all other objects (background) can be excluded from further analysis. This approach performs well in cluttered and dynamic environments and is clearly insensitive to the colour of clothes worn, even if these colours are similar to background colours.

Upper body gestures are recognised using a model based approach which is less sensitive to noise than model-free alternatives and provides a comprehensive understanding of all major joint angles (finger joints have not been tackled). The degree of freedom of the required model depends on the specific application and a trade-off between accuracy and the number of parameters of the model to be determined (Gavrila, 1999) must be made. Superquadrics are used for our body models.

Whilst other researchers have used multiple cameras from a variety of surrounding viewpoints, this approach does not suit our application where the robot carries the gesture recognition sensors and directs them at selected humans. Thus we are constrained to single viewpoint methods. We use both silhouette and depth information from a single viewpoint to extract upper torso 3D joint angles by finding the best matches between the observed data and body models. Figure 7 shows recognition results when the user's arm moves forwards and backwards. This example shows that the system can deal with self-occlusion problems.



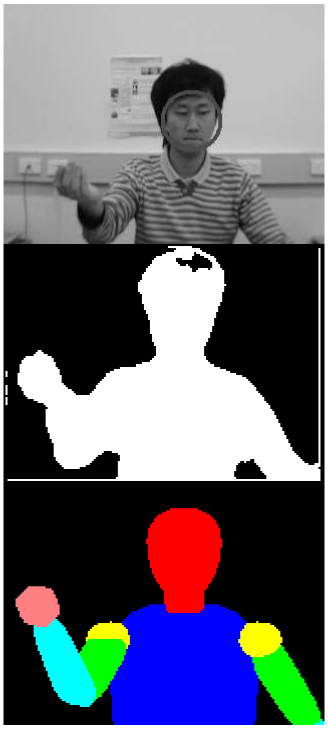


Figure 7. Image, Silhouette and Gesture Model Sequence.

The gaze direction of a person's attention is an important cue regarding nearby objects with which he/she is indicating. People tend to look at the target in the early stages of a hand pointing gesture. Using a FaceLAB (from SeeingMachines) system manufactured by Seeing Machines, which uses a pair of video cameras to stereoscopically determine head pose and eye gaze direction, we can determine a vector of the users gaze direction. The head pose and eye-gaze data can be combined using confidence-weighted factors to estimate the visual attention vector. Intersecting this vector with a hand/arm pointing_gesture and assist in identifying a target object.

So far we have concentrated on static gestures, but will soon extend the systems to understand dynamic gesturers, most likely by combining HMM (Hidden Markov Models) and FSM (Finite State Machines).

Since different particular gestures for a given intention are behavioural aspects for certain races, ethnic groups and individuals, we will customise our gesture recognition system for each registered individual who can be identified using face recognition. For those not registered or identified a default set of general gesture parameters will be used but with a lesser expectation of success.

We hope to eventually, not only combine gesture and speech but also temporal and individual behavioural cues to attempt to correctly interpret the human intention of a robot command. Furthermore we intend to permit the robot to query ambiguous commands to refine its expectations of completing a mission successfully.

Using the high fidelity 3D/colour scan of the working environment (using a Riegl LMS Z420i laser/camera scanner, as mentioned earlier) we will be able to label objects with functional attributes and to build up probabilistic models of where various objects might be found. We will then be able to design optimal search paths for the robot to find the target object with the least expected time of discovery. Behaviour patterns of individuals can also be learned and fed into the system to enhance both the Spatial Intelligence and Transactional Intelligence of the system.

We plan experiments to demonstrate the following kinds of supporting service to aged and/or fragile or otherwise disabled users:

(a) Find and fetch common objects such as spectacles, books, mugs, utensils etc.

(b) Reminders and dispensation of medications

(c) Identify users (and maintain the continuity of a transaction).

(d) Check visitor's identity at the door

(e) Lead a user to a nominated place (eg. dining hall).

(f) Track and follow humans.

The particular behaviour of characteristics of the users, time of day, known special circumstances (eg. anticipating a visitor) would be accommodated by the multi modal transactional intelligence system and appropriate clarification queries prompted by the nature of unresolved ambiguities designed to minimise the nuisance value of such. Eventually it is hoped to have a learning system adaptively refine the whole system to gradually improve the efficiency of the service processes and minimise the annoyance of clarification queries. A lot has still to be done but the framework is now in place and a number of individual components ready for integration.

Conclusions

This paper has briefly outlined the framework of a multi-model spatial and transactional intelligence system directed at having robots help aged, fragile or otherwise disabled people cope with their living environments in an assistive technology context. Much has yet to be done but the way forward is clear though complex. Only actual physical demonstration will eventually prove the system functionally feasible and worthwhile. Such is the overall goal of the project. Much is yet to be done, but the elements for an integrated system are well advanced. However, it is likely that integration will itself be a very difficult process and one which must be planned carefully, each member of the research team being aware of the need to provide clear interface data exchanges between the components in a unambiguous format devised in concert. As the project progresses we would not be surprised to find that the natural ambiguity of human to human communication will have to be resolved to an extent not initially envisioned as a pre-requirement for effective human-robot collaboration.

Acknowledgment

This research is supported by the Australian Research Council through a Discovery Grant (DP0878195).

References

[1] Axnick, K.B.J. and Jarvis, R.A. (2005), Face and pose recognition for robotic surveillance, Proc. 2005 Australasian Conference on Robotics and Automation, 5th to7thDec., Sydney, pp. 1-9.

[2] Bühler, D., Minker, W., Huluser, J and Kruger, S. (2002) Flexible Multimodal Human-machine Interaction in Mobile Environments, 7th International Conference on Spoken Language Processing, Sept. 16-20, Denver, Colorado, USA.

[3] Carrozza, M.C., Massa, B., Micera, S.,Zecca, M. and Dario, P. (2001) A ‘Wearable’ Artificial Hand for Prosthetics and Humanoid Robotics, Proc. 2001 IEEE-RAS International Conference on Humanoid Robots.

[4] Chakravarty, P. and Jarvis, R.A. (2006). Panoramic Vision and Laser Range Finder Fusion for Multiple Person Tracking, accepted for presentation at the International Conference on Intelligent Robots and Systems, Oct. 9-15, Beijing, China.

[5] Dautenhahn, K., Walters, M., Woods, S., Koay, K.L., Nehaniv, C.L., Sisbot, A. and Simeon, T.(2006) How may I serve you?: a robot companion approaching a seated person in a helping context, ACM/IEEE International Conference on Human-Robot Interaction, Proc. 1st ACM/SIGART conference on Human-robot interaction, Salt Lake City, USA, pp. 172-2006.

[6] Durrant-Whyte, H. F. and Guivant, J. (2000) Simultaneous localization and map building using natural features in outdoor environments, Intelligent Autonomous Systems 6. Vol 1, pp 581-588, IAS-6, Italy, 25-28 July.

[7] Gao, T., Greenspan, D. Welsh, M., Juang, R.R. and Alm, (2005)Vital Signs Monitoring and Patient Tracking Over a Wireless Network, Proc. 27th Annual International Conference of the IEEE EMBS, Shanghai, Sept.

[8] Gavrila, D.M. (1999) The Visual Analysis of Human Movement. A Survey, Computer Vision and Image Understanding.

[9] Hager, G.D., Chang, W-C., and Morse, A.S.(1995) Robot Hand-Eye Coordination Based on Stereo Vision, IEEE Control Systems Magazine, Vol.15. pp. 30-39.

[10] Harte, E. and Jarvis, R.A. (2007). Multimodal Human-Robot Interaction in an Assistive Technology Context using Environmental Knowledge, accepted for presentation at the Australasian Conference on Robotics and Automation , 10th to 12th. Dec. Brisbane, Australia.

[11] Hu, H.H.,Jia, P.,Lu, T. and Yuan, K. (2007) Head gesture recognition for hands-free control of an intelligent wheelchair,Industial Automation: An International Jopurnal, Vol. 34, No. !, pp. 60-68.

[12] Jafari, S and Jarvis, R. (2005). Robotic Hand Eye Coordination from Observation to implementation, International Journal of Hybrid Intelligent Systems (IJHIS), 2, pp. 269-293.

[13] Jarvis, R.A.(1985) Collision-Free Trajectory Planning Using Distance Transforms, Proc. National Conference and Exhibition on Robotics, Melbourne, 20-24th Aug. 1984, also in Mechanical Engineering Transactions, Journal of the Institution of Engineers, Vol.ME10, No.3, Sept. pp. 187-191.

[14] Jarvis, R.A. and Byrne, J.C.(1987) An Automated Guided Vehicle with Map Building and Path Finding Capabilities, invited paper, Fourth International Symposium of Robotics Research, University of California at Santa Cruz, 9-14 Aug. pp. 155-162.

[15] Jarvis, R.A. (1994). On Distance Transform Based Collision-Free Path Planning for Robot Navigation in Known, Unknown and Time-Varying Environments, invited chapter for a book entitled 'Advanced Mobile Robots' edited by Professor Yuan F. Zang, World Scientific Publishing Co. Pty. Ltd., pp. 3-31.

[16] Jarvis, R.A. (1997). Etherbot - An Autonomous Mobile Robot on a Local Area Network Radio Tether, Proc. Fifth International Symposium on Experimental Robotics, Barcelona, Catalonia, June 15-18, pp. 151-163.

[17] Jarvis, R.A. (2000). A User Adaptive Semi-Autonomous All-Terrain Robotic Wheelchair, 6th International Conference on Intelligent Autonomous Systems, July 25-27, Venice, Italy, pp. 671-678.

[18] Jarvis, R.A. (2002). A Go Where you Look Tele-Autonomous Rough Terrain Mobile Robot, 8th International Symposium on Experimental Robotics (ISER’) 2, Sant’Angelo d’Ischia, Italy, 8-11 July.

[19] Jarvis, R.A., Ho, Nghia and Byrne, J.B.(2007). Autonomous Robot navigation in Cyber and Real Worlds, CyberWorlds 2007, Hanover, Germany, Oct. 24th to 27th, pp. 66-73.

[20] Jarvis, R. A.(2009) Multimodal Robot/Human Interaction In An Assistive Technology Context, The 2nd International Conference on Advances in Computer-human Interaction,(ACHI 2009), Cancun, Mexico,1st to 6th Feb.

[21] Kanade, T., Collins, R., Lipton, A., Anandan, P., Burt, P. and Wixson, L. (1997) Cooperative Multi-Sensor Video Surveillance, Proc. 1997 DARPA Image Understanding Workshop, May, pp. 2-10.

[22] Lee, T. and Mihailidis, A. (2005) An Intelligent emergency response system; preliminary development and resting of automated fall detection, J. Telemed Telecare, Vol 11, pp. 194-198.

[23] Lienhart, R., Maydt, J. (2002). An extended set of Haar-like features for rapid object detection, Proceedings of 2002 International Conference on Image Processing, Vol. 1, pp 900-903.

[24] Lienhart, R., Kuranov, E. and Pisarevsky, V. (2003) Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection, in DAGM 25th Pattern Recognition Symposium.

[25] Lowe, David G. (1999) Object recognition from local scale-invariant features, *International Conference on Computer Vision*, Corfu, Greece (September), pp. 1150-1157.

[26] Pons, J.J, Rocon, E., Ceres, R., Reynaerts, D., Saro, B., Levin, S. and Van Moorleghem, W. (2004) The MANUS_HAND. Dextrous Robotics Upper Limb Prosthesis: Mechanical and Manipulation Aspects, Autonomous Robots, 16, pp. 143-163.

[27] Prior, S.D. (1990) An Electric Wheelchair Mounted Robotic ARM-A Survey of Potential Users, Journal of Medical Engineering and Technology, Vol. 14, Issue 4, July, pp.143-154.

[28] Rawlinson, D. and Jarvis, R.A. (2007) Topologically-directed navigation, Robotica: International Journal of Information, Education and Research in Robotics and Artificial Intelligence (on-line publication 10th Sept., (hard copy journal version imminent).

[29] Tamura, T., Yonemitsu, S., Itoh, A., Oikawa, D., Kawakami, A., Higashi, Y., Fujimooto, T. and Nakajima, K. (2004) Is an Entertainment Robot Useful in the Care of Elderly People With Severe Dementia?, The Journals of Gerontology Series A: Biological Sciences and Medical Sciencews, 59, M83-M85.

[30] Volpe. B.T., Huerta, P.T., Zipse, J.L., Rykman, A., Edwards, D., Dipietro, L., Hogan, N. and Krebs, H.I. (2009) Robotic Devices as Therapeutic and Diagnostic Toots for Stroke Recovery, Arch Neurol. Vol. 66, No. 9, pp. 1086-1090.

Utilizing Open Content for Higher-Layered Rich Client Applications

Monika Steinberg, Jürgen Brehm
Institute of Systems Engineering - System and Computer Architecture
Hannover, Germany
[steinberg, brehm]@sra.uni-hannover.de

***Abstract* - Accomplishing user interoperation and standardized web techniques is a promising mixture to build a next generation of web applications in the currently arising Social Semantic Web. Increasing heterogeneous Open Content is an ongoing trend. Generic concepts for higher-layered reuse of the arbitrary information overload - mentioning the Internet of Services - are not covered very well yet. For further, directed use of distributed services and sources, inquiry, interlinking, analysis, machine- and human-interpretable representation are as essential as lightweight user-oriented interoperation and competency in handling. In the following we introduce the *qKAI application framework* (qualifying Knowledge Acquisition and Inquiry) [1] - a service-oriented, generic and hybrid approach combining knowledge related offers for convenient reuse and tweaking them with interaction for improved access. qKAI aims at closing some residual gaps between the "*sophisticated*" Semantic Web and "*hands-on*" Web 2.0 enabling loose-coupled knowledge and information services focused on knowledge life cycles, learning aspects and rich user experience. Overall, in qKAI Open Content is boosted as an inherent part of higher-layered, lightweight applications in knowledge and information transfer via standard tasks of knowledge engineering and augmented user interaction. We introduce the *qKAI hybrid data layer* - a minimalistic data model with maximized depth - implementation results and some lessons learnt. We discuss the Semantic Web query language SPARQL critically to enlighten its limitations in current web application practice. Acquiring resources and discovering the Web of Data is a massively multithreading part of the qKAI hybrid data layer which serves as basis for further knowledge based tasks. Built upon this data layer, social educational gaming is instanced to simplify interoperation, to spread knowledge in a handy way and to enhance users' collaboration with Open Content. Attendance is increased through game-based, incentive arrangements following *Rich Client* paradigms. *Long-term objective* is to establish Open Content in information and knowledge transfer as utilized knowledge base.**

Keywords: Open Content; Social Semantic Web; Knowledge Engineering; Rich Clients.

I. INTRODUCTION

Currently the borders between Semantic Web and Web 2.0 become fluid more and more and let us create new synergies in the Web 3.0 [2] or also called the Social Semantic Web. The combination of social user involvement, employing desktop-alike rich interfaces (RIA [3]), and the Semantic Web with technologically oriented operability for data representation and processing is a promising conceptual basis to solve two pending problems. On the one side, there is still a lack of lightweight user participation in Semantic Web contexts because of handling hurdles and missing fancy interoperation ability. On the other side, there are claims for less trivial and more unitary content in Web 2.0 contexts. Currently DBpedia [4] and Freebase [5] start bringing these efforts together by offering collaborative content collection, creation, refinement, or semantic interlinking to increase Open Data that is well interpretable by humans and machines. Twine [6] is another semantic knowledge-enhancing platform, but does not offer its content with open access yet.

Metadata is an important factor for analyzing and categorizing content. In case of missing metadata, automated and manual annotations are approved workarounds to get information about the information while deriving useful knowledge out of it. Conclusions about information quality (e.g., provenance, reputation, timeliness, correctness) are important for further deployment in knowledge transfer scenarios and can be deduced out of metadata analyses and further interactive assessment.

We have to develop enjoyable interoperation scenarios that permit interactive knowledge transfer and learning. We see facilitating access to Open Data by intuitive learning interaction concepts as a promising combination to increase Open Knowledge and to prepare it for further learning purpose. Knowledge transfer is in contrast to learning a non-linear process. Learners are able to move free in the created environment and may decide on their own which learning order to take. Further on users are embedded and actively involved to influence learning sequences. Proved learning concepts have to be active, self-controlled, constructive, situative and social following successful didactic concerns [7]. Systematically linear and non-linear learning scenarios will be realized in the qKAI project [1] to allow different interaction types like exploring, questioning, answering or annotating.

Also fundamental are incentive and motivation of the users to interoperation and collaboration. Next to common methods for annotating and exploring data, using questionnaires and data browsers, we see especially knowledge games as motivating way to implicitly inquire, analyze and annotate content while knowledge is interceded. Well-designed gaming flows can impart handling of suspenseful information in an easy understandable manner to the user. Open Knowledge, that is well comprehensible for its users and machine-readable, increases this way. Newly developed learning interaction services and enriched content should be tied up with conventional Learning Management

Systems (LMS) and learning standards (LOM, SCORM, IMS/QTI [8]).

Obviously, there are many different tasks to perform, to utilize arbitrary available Open Data for higher-level, extensible and standardized applications with rich interoperation for knowledge transfer and learning. Our research showed that there are several knowledge bases, services and software components available that are required for sub tasks of qKAI. Therefore, the challenge is to merge and expand existing APIs, frameworks, autonomous services and distributed sources to perform our jobs here. According to C. Schroth and T. Janner [9] we see the relation of our needs to service-oriented software design (SOA): "*The first major analogy between product design in the fields of Web 2.0 and SOA is the notion of reusing and composing existing resources. Both concepts let users reuse, remix, and enrich existing resources and components to new and potentially higher-level applications. The second commonness is the affinity to collaboration and coupling of remote resources or services. Both Web 2.0 and SOA applications enable the loose coupling of distant and possibly heterogeneous resources. A third apparent resemblance between Web 2.0 and SOA is the shared principle of agility and the support of permanent structural change.*" [9]

Long-term objective is to embed different types of services (atomic, simple and composite services) in qKAI for systematically utilizing Open Data and enhancing Open Knowledge. Design concepts from service-oriented and mediator-wrapper-based information systems [10] are applied in the system specification of the qKAI framework. We identified three main service categories and packaged them in three service bundles as interaction, representation and discovery manager in a mediation layer (see Figure 2). To keep the system structure comprehensive and easy extensible we take a 4-tier-layer concept paired with Rich Client MVC2 paradigms to structure and model desired service managers and types.

A. Structure of this contribution

First, we introduce some further background. In Section 2 follows what we see as prerequisite to utilize Open Content for higher-layered applications. Section 3 gives an overview of the qKAI application frameworks' system design. Section 4 offers some more details concerning the qKAI hybrid data layer as one system level of the 4-tier design. Section 5 shows further services and components, Section 6 exemplifies use cases and further application scenarios. At least this contribution ends up with a conclusion and future work in Section 7.

B. Resources and Open Content

Open Content is interpreted in qKAI following the Open Knowledge specification "*Defining the Open in Open Data, Open Content and Open Information*" by the Open Knowledge Foundation [11]: "*A piece of knowledge is open if you are free to use, reuse, and redistribute it.*" qKAI adds processing differentiation between Open Data, as raw input information and Open Knowledge, which represents qualified information – checked or enriched yet. The semantic Web of Data (RDF stores and ontologies) and User Generated Content (Wikis, communities, Blogs) stand by and grow up in structured, unstructured and semi-structured manner. DBpedia offers an extensive knowledge base in RDF format [12] (generated out of Wikipedia content), allows semantic browsing and detailed thematic inquiring by applying SPARQL [13] queries for refinishing and further assignment.

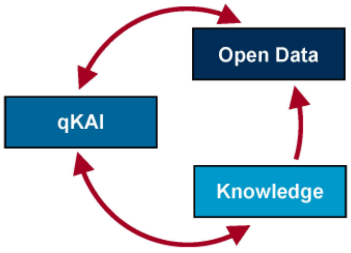


Figure 1. qKAI conept scheme to derive knowledge out of Open Data

RDF aims at the unique description of entities, their relations and properties on the internet [14] according to a standardized schema. These are the resources or "*things*" if we talk about the renewed "*Internet of Things*". The access to resources is always carried out using representations. One resource can have several representations like HTML, RDF, XML or JSON.

Open, shared databases like Freebase offer a free API to reuse its content by its own Metaweb query language (MQL) [5]. Relational databases can be easily converted into the Web of Data embedding existing components like the D2R server [15]. Additionally, many unstructured sources like HTML sites or PDF files do not apply to machine-interpretable web concepts yet. Serializing this data to standardized formats with open access is a first step towards enhanced machine and user interpretability. Aperture [16] and Virtuoso Spongers [17], for example, enable comprehensive solutions for these tasks. In case if more text engineering is needed, there are comprehensive solutions for standard Natural Language Processing (NLP) tasks (e.g., by OpenNLP [18]) to perform sentence detection, NER (Named Entity Recognition), POS (Part-Of-Speech) tagging or even semantic chunking.

C. Quality of content

Especially, if enabling User Generated Content (UGC) that is not authored by experts yet for knowledge transfer scenarios, the question of contents' quality arises. Therefore, we developed a **three level model** to handle different aspects of quality. Metadata can be seen as a quality feature [19]. The more metadata we are snapping, the better we get to know the content. There is no absolute quality, but we can compare resources with each other (Open World Assumption) and weight them based on the amount and structure of metainformation. Enrichment of a resource happens in a corresponding qKAI URI by semantic interlinking. One example is a domain ranking visualized as tag clouds to show from which domain we get the most information right now. First level criteria contain metadata directly included in a resource like format, timeliness, author, provenance or language, which can be automatically detected. Second level criteria are determined through user interaction, which helps to enhance semantically correctness. Regarding factual knowledge like "*Berlin lies at the Spree*" or "*Hanover is the capital of Lower Saxony*", we see user rating and ranking following the established Web 2.0 manner as an effective solution to mark wrong content and to rank valuable or popular content systematically. Next to this crowd sourcing community approach, we offer role- and level-based quality control mechanisms. Lecturers earn rewards while rating and creating educational resources out of Open Content; students earn rewards while answering questions, managing gaming tasks, exploring further content or ranking their favorites. Gradually content can be qualified this way. Resources are marked following their quality level as reviewed, proofed or not yet qualified to enable embedding in different levels of knowledge transfer and learning. Third level criteria are inferred employing Natural Language Processing to detect some more information hidden inside a resource.

D. Interaction in the Social Semantic Web

We are looking for some innovative interaction to deploy Open Content for further purpose. The state of the art shows for example DBpedia mobile [20] that combines Wikipedia entities with images and places them on a map. At Revyu [21], we can rate and rank everything. For selection and requesting Open Data in RDF format there are SPARQL query tools available. Most of them are not very intuitive but technical. Altogether, we currently can **search, display, group, edit and annotate** content on the web. However, there is little cumulative advantage and not much incentive for the user to interact and to deduce new knowledge.

E. Games with a purpose, social and educational gaming

The idea of combining social networking, gaming and rating is not new. As far as we know, there are no applications available adding strong focus on knowledge and learning to it. Available social games do not rely on standardized Open Content or sustainable concepts. Gaming content is explicitly and laboratory created manually for every single game. Generic approaches to build an ongoing social knowledge network based on available content are still missing. Embedding gaming into superior learning structures regarding learning management standards, e-learning infrastructures and the Internet of services seems to be not mentioned so far. Different to other gaming approaches content creation in itself is part of our game-based learning concept. Players get the ability to change the view of relevant resources. For example, text and multimedia is presented as extracted chunks of information or image having knowledge-snack concepts on mind to enhance understanding and portioning not to overburden the user. Text sections out of articles are passed to the user and he has to guess the context. Sights are presented in detailed zoom view to let users guess them. Locations have to be placed at the right position on a map.

Luis van Ahn introduced crowd sourcing to gaming with ESP game or reCAPTCHA [22]. Guess-the-Google is a term-guessing game based on Google search results or images [23]. Scoyo [24] offers a game-based learning platform for kids, but does not deal with Open Content or Open Access. All over, there are no generic game-based learning concepts available regarding web standards and learning management based on Open Content. Additionally there are some commercial, social gaming applications like Playfish, MegaZebra or Zynga [25] with casual character. They are often embedded into Web 2.0 platforms like Facebook or MySpace [26] to increase participation.

Open Content is a huge knowledge base, but there are missing augmented interaction abilities to confront users with Open Knowledge bit by bit in enjoyable manner (knowledge-snacks, casual games). We can read Wikipedia articles or watch some educational videos at e.g., YouTube, but knowledge-centered approaches reusing available content in a standardized and generic manner (domain independent) are still missing. We are looking for mechanisms that bring more motivation and incentive to the user while interoperating with Open Content. Therefore, we chose a game-based learning approach embedding Open Content in knowledge games. Assessment methods - e.g., self-assessment during lectures - to hold students' attention integrated in Learning Management Systems (LMS) like ILIAS [27] showed good evaluation results among the students and they asked for more quiz-like interaction [28]. About 70 percent of the students did the quizzes by themselves at home again to repeat the material. Workload to prepare and realize online assessment and quizzes is very high – so we are searching for (semi)automated approaches to generate e.g., question-answer-pairs. Furthermore, the web offers a lot of information and knowledge available as Wikis, semantic or geographic data and multimedia.

F. Social Semantic Web behind the scenes

RDF: The Semantic Web, increasingly transcribed as „*Linked Data*" [29] or "*the Web of Data*", is supposed to bring new quality to the internet: What was formerly known in form of internet pages only to human beings, shall now be applied to automatic processing. In order to achieve this, the formerly in continuous text existent data will be classified, its properties transformed into defined forms and, as their aggregation, connected through labeled links. The schema of the „*Resource Description Framework*" (RDF) [12] - developed for this purpose - follows a natural speaking sentence structure. It consists out of the following information carrier: The "*subject*", "*resource*" or "*node*" is presented as an URI (Unified Resource Identifier) just like the "*predicate*" and the "*object*". All of them might contain properties following their description. Properties their self are typed and if we imagine RDF as a tree, they represent the leaves. Their type is normally declared like for example the number 42 is of the type "*integer*", but functionally dependent from its predicate. The relation of the information carriers is modeled implicitly, always directed and qualified through the predicate. Instead of speaking about "*subject*", „*predicate*" and „*object*" (the object might be a subject as well), it is more efficient to name them "*properties*" that are assigned to resources. Resources are connected in three ways over relations: As source, target and identifier.

SPARQL: With SPARQL (SPARQL Protocol and RDF Query Language) [13] a search and query language for RDF repositories is designed. SPARQL is a W3C specification since the beginning of 2008. SPARQL's syntax is similar to SQL while columns can be defined to answer requests. Filtering expressions are possible in SPARQL that are placed in the WHERE clause in SQL for example. Actually, there is no efficient functionality to implement full text search yet. Next to ASK there are no aggregate functions available in the SPARQL specification at this time. ASK allows only a true/false statement about whether a request delivers a result or not. Abandon the special case of an identity, regular expressions should be used for full text search. Such expressions do not fit properly to a large amount of data, because up to now there are no database indices available to speed up them upon text. That is why every expression has to be evaluated for any RDF property and all of the properties have to be fully loaded too. To get more aggregate functionality next to ASK, many providers implement additionally, proprietary extensions. This up to now not standardized extensions use the strength of the traditional, relational query language SQL and a combination of SPARQL and SQL. For this reasons also qKAI does not use SPARQL only. Temporary query results have to be stored anyway, to allow acceptable performance while requesting and combining distributed RDF resources. These results are stored in a relational database – MySQL – and SQL is utilized for effective, internal data processing (see Section 4).

REST: Representational State Transfer (REST) [30] [31] is an architectural style - not tied to any particular technology - although it is used as a guide for designing architectures following four key constraints: identification of resources handled by URIs, manipulation of resources through representations using the HTTP's uniform protocol (GET, PUT, POST, DELETE), self-descriptive messages and hypermedia as the engine of application state. REST was defined by Roy Fielding [30] in his dissertation as an architectural style he used for foundational Web specifications - in particular HTTP and URIs. REST offers important architectural properties to improve reliability, scalability or simplicity. These properties are often named as superior to SOAP web services so that we can speak about REST as a "*thin*" style SOA alternative. Especially in Web 2.0 applications, REST web services are very popular these days.

Rich Clients: Regarding today's browser-based user interfaces, Rich Clients using especially asynchronous JavaScript and XML (AJAX) are a wide spread trend. The aim of Rich Internet Applications (RIA) is to bring desktop-alike and easy to use interoperation to the Web. Next to AJAX, Adobe Flash/Flex [32] and Java based User Interfaces (UI) are technical alternatives. The best technique to choose depends on the main requirements that have to be fulfilled [3]. Flash/Flex3 for example offers the advantage of less scripting work and easier event handling to reach highly interactive functionality, if the focus lies on multimedia and design issues. All these Rich Clients can be seen as an extended and enhanced view in the traditionally Model-View-Controller (MVC2) concept. Advantages Rich Clients offer are e.g., faster reaction to user requests with partially reloads of site parts without refreshing the whole site, less network traffic and server load as well as offline usage possibility. A so-called Rich UI Engine delivers the General User Interface and the presentation logic is divided from visualization components. RIAs as standalone Web clients that interact with the server side through web services are a promising combination. One of the most important advantages of the Client-Server model is the idea that the User Interface should be developed independently of the business logic and data persistence technology. Nevertheless, to be honest, in today's Web programming practice before RIA, the UI is in fact tightly coupled with the server-side technology of choice. If we like to change our backend functionality for example from Java to PHP we also have to rework all the scripts generating the HTML UI from *.jsp to *.php. To avoid this problem we can now choose a Rich Client communicating over standardized interfaces like web services only and put the most of the UI logic to client side to get a real separated solution.

II. HOW TO UTILIZE OPEN CONTENT FOR HIGHER-LAYERED APPLICATIONS?

In this section, we outline what we see as requisite to turn Open Content into an organized, useful knowledge base for higher-level applications. We are aiming at the establishment of powerful, but by the user easy to handle mechanisms for acquisition and inquiry of relevant data out of heterogeneous sources. We have to serialize formats for unitary, comprehensive analysis and mediation of distributed, inconsistent content. Mediation means here to utilize input data for higher layered applications by offering personalized query plans, transformation, annotation and interoperation. Open access to knowledge and data in e.g., RDF representation brings advantages in interlinking and easily accessing distributed data on the Web. Data processing concepts allowing machine- and human-interpretable staging without storing redundant data permanently become possible by semantic interlinking.

Information chunking for easy to digest knowledge bits without losing context information is needed for better understanding and human-capable representation during interaction. In qKAI (semi-)automatic extracted information units represent a piece of information that can be qualified by annotation and interaction to a knowledge unit – always aware of its context not to lose information and to allow effective change management (actuality).

Knowledge life cycle concerns have to be matched with content cycles of the Read-Write-Web. Acquiring (inquire, discover, categorize, index) maintaining and mediating (manage, analyze, enrich, transform) and particularly reusing (interoperation for information, learning, knowledge transfer) services have to be established.

The more we know about a source, the better we can reuse it. Therefore, metadata and its annotation are essential for accurate thematic, semantic analysis and quality determination. Determining the quality of content enables us to rearrange according to source criterions like provenance, timeliness or correctness. Emerging qualitative valence of information units and sources raises information to valid knowledge. To get the emerging qKAI knowledge base applicable, interaction services for learning, rating, ranking, inquiring, exploring and annotating are needed. Motivation and user involvement are important aspects and learning games are proficient for easily accessible, intuitive forms of interactivity. Synergy effects between learning, gaming and annotating content arise. Content enrichment by the user is seen as an implicit, positive side effect in qKAI application services.

Learning scenarios in qKAI enable self-controlled and directed concepts embedded as interaction services. We are starting with the scenarios shortly outlined in Section 6, but extension is kept simple by architectural division of presentation, its logic and managing interaction services. Learning targets deal with competency in information handling and learning in joyable manner with the Internet of Services.

RESTful SOA (Service Oriented Architecture) paradigms harmonize with Web 2.0 concerns. They support Semantic Web technologies as scalable, reusable and unified software concept, while retaining application autonomy and put the resource in the center.

III. State of the Art and Related Work

Currently some novel Social Semantic Web applications (e.g., Twine [6], Freebase [5], Knol [33]) arise that regroup existing knowledge, allow manual annotation and creation. There is still a lack in all-embracing, standardized frameworks, integration practice and reusing interoperation scenarios. A few examples for game-based interaction with Open Data like Quizzer [34] are available - embedding User Generated Content or annotating the Web of Data. Interaction and learning applications that combine arbitrary sources in an extensible SOA way are not available yet, as far as we know. DBpedia mobile [20] is an ideal example for browsable linked data combined out of different sources on mobile devices, but interactive learning scenarios, change management and further web service integration have still to be applied. SOA concepts find more and more their way into university and campus management systems. New qKAI services can be loosely coupled and integrated. Frameworks helpful for the GUI side use Asynchronous JavaScript and XML (AJAX: Prototype [35], Dojo [36], YUI [37]) or Adobe Flash/Flex [32] (e.g., FlowUI Open Source RIA Enterprise Mashup Framework). The OpenRDF Sesame framework [38] brings comprehensive JAVA functionality for semantic data discovery, querying and transformation. The Semantic Web Application Framework [39] enables further semantic tasks.

Semantic search engines like Swoogle [40], Sindice [41] or Watson [42] deal with searching, exploitation and large scale access to the Semantic Web of Data and can be used as gateway to find further relevant input resources for the qKAI hybrid data store. Nevertheless, beyond getting to know where to find these resources, qKAI wants to transform and embed the resources' content after mediation in own application scenarios. Therefore, we have to add its own open and transparent data processing, storage concept, representation and change management that is outlined in Section 4. To crawl the Web of Data we can use available solutions, but for effective storage, recombination and real time access of the resources' content during user interaction we developed the qKAI hybrid data layer. Additionally, existing crawlers are pluggable into the qKAI data layer.

IV. qKAI Application Framework and System Layer

The system design of the qKAI application framework is organized in four main layers as a combination of mediator-wrapper-concepts [10], service oriented approaches (Internet of Services) and conventional web application N-tier design. In this section, we explain the components and tasks of the applied layers as shown in Figure 2.

The presentation layer implements the General User Interfaces and its necessary logic. To fulfill extended MVC2 separation, the mediation layer presents the business logic and controller functionality. Regarded in a SOAP style, service-oriented way we would place the Enterprise Service Bus (ESB) here and the service broker belonging to the discovery manager. The mediation layer acts as middleware that connects available services (service mediation) and other technical components. The definition of "*mediation*" in qKAI is also interpreted according to Wiederhold [10]: „*A mediator is a software module that exploits encoded knowledge about certain sets or subsets of data to create information for a higher layer of applications.*"

The data layer meets the model level in the Model View Controller pattern and extends it with wrapper services at the

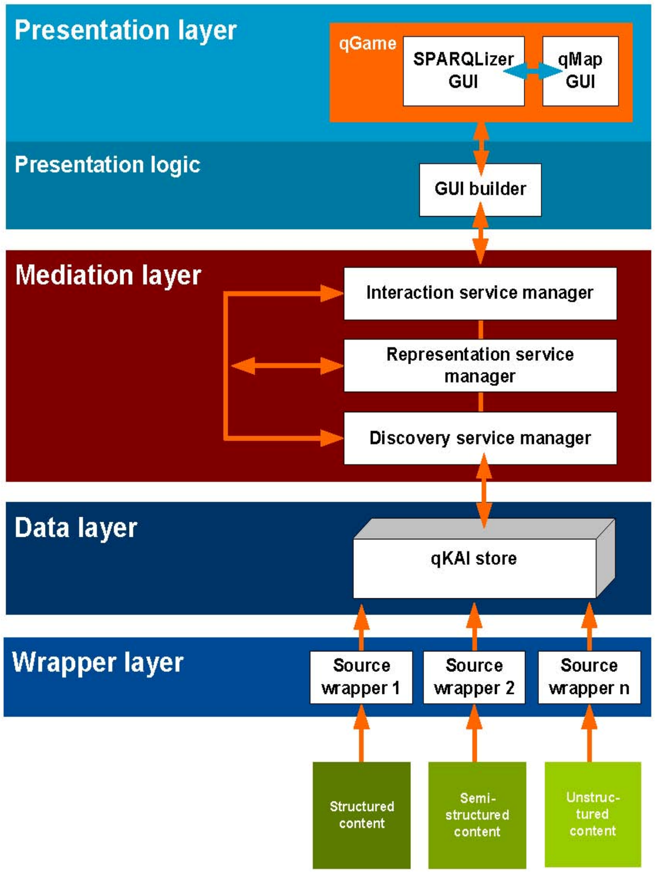


Figure 2. qKAI application framework: System layers as conceptual design and development basis

wrapper layer to embed various distributed sources. The data layer has to manage hybrid data processing enabling RDF and XML related data as well as relational database content. Existing data sources are temporarily retained for mediation purpose. qKAI processes provide new generated knowledge as open RDF or serialized JSON representation after mediation.

V. The qKAI hybrid data layer: Some details

The qKAI knowledge representation consists for example of RDF graphs and superior metadata about them. Existing graphs outside of qKAI are first stored as link to the origin source in the qKAI service and source repository e.g., using the URI of a SPARQL endpoint like http://DBpedia.org/sparql. New generated information is stored separately and sustainable at the qKAI data layer. The data processing concept contains a persistent, relational database component, flexible representation (RDF, N3, JSON) and temporary fetching of working data during discovery and enrichment. To allow persistent storage and to handle relational data with only a small set of classes in an effective way we deployed the Java persistence API (JPA) with the EclipseLink implementation [43] and Builder patterns.

Linked Data concepts [29] enable persistent, resource-related storage, change management and non redundant, context-aware data processing by interlinking identifiable distributed resources. By qKAI services generated knowledge about available sources is stored additionally in MySQL and can be represented as Linked Data on demand. Figure 3 illustrates, how the qKAI data layer embeds Linked Open Data resources and represents its data store as another node in the Linked Open Data cloud. Following the Linked Data keynote and overall desire of next generation web applications, to combine different, distributed RDF stores on demand in real time without buffering, our first trials to embed content while querying exemplary the DBpedia endpoint, failed. Without fetching a copy of relevant data, results showed that bad response times that we had to develop a more practically solution. Main reason for the bad response times is the SPARQL endpoints performance as outlined in the following (see Section "*Fetching and indexing*"). Now the qKAI data layer buffers relevant resources in a traditional, relational database structure to allow adequate performance to users' requests and ability to further processing and annotation of the acquired content. Further on, affordable hardware can be used to reach good performance results without the need to investigate in high-end enterprise server technology.

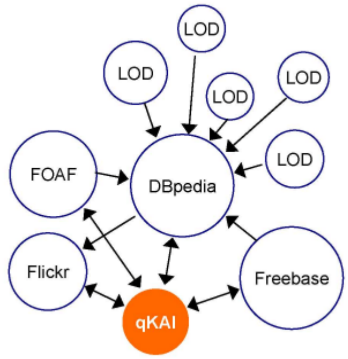


Figure 3. qKAI and the LOD cloud

This section gives an introduction into the qKAI data management based on the concept that every RDF resource is portable into a relational database structure and vice versa. DBpedia is used as an exemplary knowledge base accessed via its SPARQL endpoint to demonstrate first results while building the hybrid qKAI data store out of distributed web resources. Main criteria for the qKAI data store are easy, affordable, scalable reusability, possibility to mediation and representation of acquired resources of different kinds and varying provenance. Points of Interest (POI) are deployed to give a starting point into the knowledge base by users or further applications and to update the qKAI store partially on demand in an effective way.

A. *Discovering Linked Data by setting Points of Interest*

Because of the wide web discovery space (open world assumption), conceivable performance problems answering, and updating comprehensive queries in real-time, among others we are integrating POI (Point of Interest) setting functionality in qKAI. Further applications and the users need an interface to define points of interest and to limit inquiry space and knowledge base according to own interests. Setting and storing several POIs according to different domains and themes becomes possible. Once set, POIs are stored for reuse and are interlinked with each other. The POIs enable defined entry points to the large RDF knowledge base and enable huge databases to update temporary redundant stored data to be efficient synchronized with their provenance source on demand when requested. Interesting parts of domains and knowledge bases will be updated even often then irrelevant ones and weighting is set this way implicitly for further statistically tasks. We implemented a thin, asynchronous updatable and practicable data layer this way. Browsing Linked Data stores requires semantic browsing functionality for different devices like web browsers or mobile devices. Representational Web services are designed to fulfill device specific representation requests of the qKAI knowledge base.

A single geographically and/or thematically oriented Point of Interest (POI) is now using the qKAI implementation completely reproducible by the qKAI data layer through a single and effective SQL query. In the current implementation, it is assumed that the topic the user is interested in is a part of the knowledge base yet – now DBpedia for testing purposes. DBpedia contains all thematically categories out of Wikipedia – so we know approximately, what we got up to now. Under this condition a thematically search space limitation is representable through the following SQL statement:

```
-- POI for persons
SELECT DISTINCT r.resource, r.resourceURI
FROM resources r
JOIN relations l ON (r.resource=l.resource)
WHERE l.predicate
= ANY (SELECT resource FROM resources WHERE
resourceURI="…/22-rdf-syntax-ns#type")
AND l.target = ANY (SELECT resource FROM
resources
WHERE
resourceURI="http://DBpedia.org/ontology/Person");
```

POIs can be combined without limitation. In SQL, this means an implementation as follows („NATURAL JOIN“ is a substitute for „INTERSECT“ which is not supported by MySQL):

```
SELECT resource FROM
(/* SELECT from POIa */)
NATURAL JOIN
(/* SELECT from POIb */)
NATURAL JOIN … -- for further POI
```

The derivation of this high-performance, multithreading solution to embed distributed Linked Open Data in higher-layered web applications is exemplary enlightened in the following. A **minimalistic data model** with **maximized depth** is applied, because the implementation is suitable for several semantic data purposes on a Quadcore hardware platform that is available for about 500 € (May 2009). Because every table can be mapped to RDF and vice versa, there is no need for further extension. Further on, relational data storage brings the advantage of flexible, independent representation and delivery on demand e.g., in JSON next to RDF/XML or N3 Turtle.

B. SQL replaces SPARQL

Considering the current infirmity of SPARQL shown in the introduction and regarding the fact that popular SPARQL implementations are still a facade for relational databases, it seems to be consequent at this time to abstain from the broad adoption of SPARQL. Significantly, even DBpedia uses Virtuoso for a SPARQL endpoint with proprietary extensions, but MySQL is used for query processing in the background. Thus, we decided to choose MySQL directly for the qKAI data layer and reduced the data processing bulk to SQL that most developers are more familiar with up to now. On the one side, we can resort to a broad pool of proofed SQL solutions – enabling hierarchical queries for example – on the other side following developers save incorporation time to get started with the qKAI data layer.

SPARQL is used only to acquire DBpedia content or other RDF stores using hybrid indexing. Our first approach, sending complex SPARQL requests without SQL background support to the endpoint, failed. Even for a tree out of more than two nodes, results often could not be returned and the endpoint timed out. Under these conditions, we discarded this approach. Present practice showed that too many dependencies on distributed resources constrain further work that relies on the qKAI data layer, because the reliability of reaching SPARQL endpoints is not well calculable this times.

C. Hybrid knowledge index

An initial qKAI knowledge base e.g., out of DBpedia can be easily imported using the qKAI dump reader. This initial knowledge base is complemented on demand by the proper connected resources via SPARQL. The dump reader accepts requests and queries the buffered qKAI data layer instead of the distant resource. A hybrid knowledge index arises this way by and by.

D. Change management

The in MySQL buffered qKAI knowledge base will become obsolete earlier or later. Thereby the qKAI data store might be soon as large that synchronizing all entries with their provenance resource is not possible anymore all at once. The hybrid knowledge index allows updating the buffered data store partially at runtime. If a user or an application signalizes that more elements out of e.g., DBpedia are needed while setting Points of Interest, they can be additionally loaded and their content can be updated on demand. If the update amount and interval is set properly, "*fluent*" actualization of the whole qKAI knowledge base is possible. Only small and relevant amounts of data have to be fetched out of the provenance source this way and they can be parallel processed in the qKAI data layer. Data can be provided reliable at any time this way.

E. Reusability and extensibility

All database connectivity of the qKAI data layer is implemented autonomic and persistent without relying on other qKAI components. To embed further SPARQL endpoints we only have to deploy the new endpoint's address. The implementation of generic classes allows using even different protocols – what we showed exemplary with the qKAI dump reader. New sets of POIS can be integrated the same easy way. At this time, we offer POI setter for thematically, geographically and full text POIs. Full text POIs search like traditional search engines for matching phrases.

F. Multithreading

Without to beware of it, most developers of web applications work with multithreading on multicores. In case of Java, the Container manager treats every request as a separate thread. Most of the time there are no negative side effects, if the result is not especially tweaked for this purpose. A different case we got with the qKAI data layer that is forced to be driven parallel to reach practicably performance. In addition, blockades while querying to slow answering resources had to be avoided.

G. Fetching and indexing

A first, not yet thread optimized version of the qKAI data layer claimed over 30 seconds to fetch some test data and all of their properties. Most of the time was wasted with the one thread of the application waiting for DBpedia answers.

To understand the underlying processes and their problems to infer a more reactive solution, we analyzed the tasks to do while fetching our test data from a SPARQL endpoint:

1. For a given word the URI has to be found.
2. All properties and relations of the constant URI and their neighborhood are fetched. This happens

gradually because SPARQL does not allow binding constants to variables.

3. Step 2 has to be repeated for every fund.

H. DBpedia Search example with the term „Nietzsche“

Now we take the famous search term „*Nietzsche*“ , looking it up in DBpedia using SPARQL only and we get the following results:

TABLE I. NECESSARY REQUESTS SEARCHING FOR "NIETZSCHE" IN DBPEDIA

Step	Document count	Requests
1	444 hits for search term „Nietzsche“	1
2	8458 properties 41804 outgoing links 2068 mutual properties among	444

Table 1 shows the necessary steps, document count and requests searching for the example term "Nietzsche" in DBpedia using SPARQL only. In general, all hits have to be loaded, because SPARQL does not know any word quantifier ordering and the wanted term might be at the end of the result list. We get a serial waiting time for 445 query results with 50.000 new created documents or even markers. This result can be constructed much shorter enabling qKAI buffering DBpedia and then applying the following SQL queries to the qKAI data layer:

```
SELECT resource
FROM properties_fulltext
WHERE q="Nietzsche"
```

Repeated for table „properties“:

```
SELECT COUNT(*)
FROM (
SELECT resource FROM properties_fulltext
WHERE q="Nietzsche"
) sub JOIN relations p ON (sub.resource=p.resource);
```

Also for "relations":

```
SELECT p.property, COUNT(*) amount
FROM (
SELECT resource
FROM properties_fulltext
WHERE q="Nietzsche"
) sub JOIN properties p ON (sub.resource=p.resource)
GROUP BY p.property HAVING amount > 1;
```

We noticed that a trivial DBpedia request is answered in 2 seconds at its best. Thus, without synchronous loading we have to wait 14 minutes at least. To solve this problem, we divided the process in a pipeline of five steps to identify especially working and waiting intensive tasks. Finally, we matched these tasks to a suitable number of threads and processor cores. In our implementation, the Java classes do not communicate directly. Instead, they use the „*FetchingMediator*" as broker. For every step represented through a class, a predefined amount of „*ThreadPoolExecutors*“ is started. Class instances are lined in a queue. Every started thread will work constantly and multiple threads accomplish tasks parallelized. With four concurrent „*RelatedResourceProducern*“ the waiting time of the above mentioned example theoretically increases to a quarter. Practically DBpedia answers slower with increasing concurrent number of request. Next to qKAI there are several other applications querying DBpedia what makes the answering times unforeseeable. Waiting times of several minutes cannot be foreclosed.

The **parallelized approach** prevents the bottleneck in qKAI: Following requests might be answered faster allowing queued search terms to be processed further on despite of small blockades.

I. Evaluation of the qKAI hybrid data layer

Initial search space creation without preselected data dumps: If data is fetched only by requesting a SPARQL endpoint like DBpedia without buffering, the used time until the whole search space is available, depends on its connection. qKAI itself uses only very simple SPARQL expressions and - if implemented in the endpoint - the Virtuoso extension "*bif:contains*" [17] to allow faster full text search. According to our samples it takes about 20 seconds until 10 results are available: To be presentable, a result has to be found and must be fully loaded as well. Resulting out of this unacceptable response times while only using SPARQL requests on distant resources, qKAI uses a prepared data store as copy of the provenance resource to utilize distributed data for further application scenarios in real time. Therefore we currently dissuade from deploying "*SPARQL only*" installations in productive environments – even if this approach is desirable regarding the long term, it is not realizable in an affordable way up to now. Instead, a minimal data store - with at best domain or application customized data - should be prepared like explained in the following using the example of the qKAI data layer.

Initial search space creation with preselected data dumps: A "*dump*" is a view of a database written down in a file. In case of DBpedia, we speak of 40 compressed files that reach decompressed 25 GB. The decompressing time took about half an hour on test system 2 (see Table 3). This data has to be transformed from N-Triples format into a relational data base structure to allow more effective processing and to optimize further queries. N-Triples are a rather redundant data structure because e.g., a RDF subject is repeated for each of its properties. Directly transferring of N-Triples into the qKAI data store is a quite ineffective task. It would mean to get tables with over 200 million lines and expensive JOINs with more than one property concurrently. The up to now fastest known SPARQL implementation Virtuoso [17] uses just like qKAI OpenRDF Sesame [38] to parse data into a more effective representation. After cleaning up, the data are transformed into the qKAI data model. Therefore we turned off the database indices in MySQL because it is a rather time intensive task too.

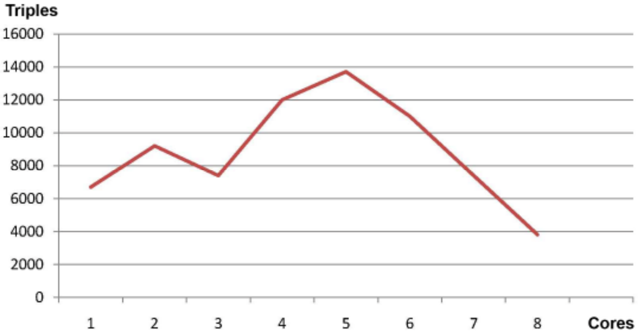


Figure 4. Count of read triples per second and amount of processors per second, measured over 80h, while reading in DBpedia dumps.

Figure 4 visualizes reading in the DBpedia dumps on test system 2 (see Table 3). The peak lies at five by qKAI used processors because there is an optimal balance reached between 5 cores for qKAI resources and 3 cores for MySQL. In average, even 4 cores fit well.

J. Result concerning the qKAI hybrid data layer

If a distant RDF store like DBpedia is mapped to the qKAI data layer as relational structure, efficient search space limitation and processing of the content is realizable with good performance using internal SQL selection. Requesting and combining distributed RDF resources using SPARQL only currently turned out to be not practicable. Therefore, we implemented the transformation into an own, efficient relational data structure. At this time, the qKAI data store consists out of DBpedia. We will enhance it with further resources by and by. The qKAI data layer can be used to embed available semantic search services to build a hybrid knowledge index for the Semantic Web - and in future work for further Open Content resources.

Embedding e.g., the Sindice and Watson crawling functionality is under development. Sindice offers a large RDF data index; qKAI is able to mediate with to acquire further resources next to DBpedia. The algorithm for change management introduced in Section 4 can be used for periodic and on demand synchronization with provenance sources. Relying on a modular architecture, qKAI can be easily extended by further Points of Interest e.g., for multimedia content.

K. Test systems

The implementation of the qKAI data layer is tested on two different systems to determine the ideal distribution strategy:

TABLE II. BENCHMARK DATA FOR TEST SYSTEM 1

Processor	AMD Phenom 9550, Quadcore, 4x 2,2 GHz
RAM	8 GB DDR2 667MHz RAM
Internet connection	14 MBit
Storage	Four Western Digital WD6400AAKS SATA 2 hard disks, Adaptec 3805 RAID Controller in RAID 5
Throughput storage	380 MB/s read, 300 MB/s write

TABLE III. BENCHMARK DATA FOR TEST SYSTEM 2 (AMAZON ELASTIC COMPUTING CLOUD 2)

Processor	2x Intel XEON, Quadcore, 8x 2,33 MHz
RAM	7 GB, unknown
Internet connection	1 GBit
Storage	Amazon EBS Storage, Western Digital WD6400AAHS
Throughput storage	60 MB/s read, 50 MB/s write

VI. SERVICES AND COMPONENTS IN QKAI

To keep the qKAI application structure flexible, extensible and autonomic, we decided to encapsulate functional subtasks in small web services. Web service interaction will follow RESTful Web 2.0 paradigms. Self-descriptive messaging is the most important constraint of REST. It means that every message needs to include all the information necessary in order to understand the message itself. We do not see a web service description language like WSDL as mandatory for REST – also it is possible to describe RESTful web services using WSDL 2. The fundamental advance of REST over the styles of e.g., SOAP or CORBA is that all service interfaces are the same. There are no differences with the need for explicit description. After registering all available services and resources in the qKAI data store according to a broker, we are looking towards embedding structured databases by converting them to RDF. The last challenge is to enhance unstructured content for qKAI integration developing more complex wrapper services.

We are dividing qKAI functionality into two main developer levels:

1. RESTful backend web services for acquiring, selecting and representing textual and multimedia data. Read and write access to resources is performed over the HTTP protocol by the GET and POST method. Working at this level means extensive use of the Java Jersey API (JAX-RS specification) to build atomic web services for several subtasks in effective manner.
2. Rich User Interface components let the user interact with different kinds of activities. Frontend components are built with AJAX and/or Flash/Flex according to their type of interactivity.

A. RESTful knowledge engineering

We deploy RESTful web services (atomic and composite) to handle standard tasks: acquire, represent, transform and annotate resources. Loosely coupling of remote resources and services becomes possible; stateless services and server communicate over the http protocol.

B. Modular rich frontend

According to Rich Internet Application (RIA) concepts, user interfaces are executed at a stateful, client side Flash runtime. Desktop-alike applications offer advantages like faster reaction to user requests, less network traffic, less server load and offline usage possibility. We decided for a Rich Thin Client using Adobe Flash and Flex - so business logic remains at server side. The Flash player acts as rich UI engine and delivers the GUI. The presentation logic is divided from visualization components. Flash is opening up its format with the Open Screen Project [44]. The Flash format and Flex SDK are not that proprietary anymore like some years ago. Nowadays the Flash plug-in is spread over 90 percent around browsers. Flash also is promoted through spreading its new FLV format in online video communities like YouTube. So Flash obtained enormous ubiquity and lost a lot of its proprietary nature these days. Browser, platform and device independent developing interactivity is the big advantage of Flash/Flex applications compared to Ajax. Flash is predestinated to design gaming content because of its effective possibilities with high design issues and focus on usability.

C. Interaction and game-based activity

As Ahn's reCAPTCHA, ESP game [22] or Amazon Mechanical Turk established gaming as a well-suited instrument to solve several tasks in knowledge engineering. However, they do not mention any learning or knowledge concerns while gaming. On the one hand, users can enrich content; on the other hand, users can learn and share knowledge through gaming with Open Content in a social incentive and challenging way. We are aggregating existing information and enriching it while interacting with Open Content. Even statements about contents' quality can be deduced out of users' content and activity. Especially fact-related knowledge can be transferred and learned, if resources are presented in rule-based manner to the user and he has to solve predefined learning tasks earning rewards.

Creating and editing: We support authors to create and combine content. For example, false answers are automatically generated by wrong-answerizer services, if a question is generated by a user. qKAI offers proposals for new learning games out of given domain knowledge and concepts. Matching between available content and suitable gaming types is described in ontology-based concepts. At the same time, the user enhances underlying ontologies while deploying game types and rating gaming content. We want to create self-organized ontologies that adaptively grow with ongoing user interaction.

Interlinking and grouping: Grouping of existing resources and interlinking with missing ones is rewarded e.g., with in-game incentives.

Rating and ranking: Instead of simple questionnaires, rewarding concepts are deployed to get user feedback.

D. Educational aspects and suitable domains

Gaming is not overall suitable to learn every skill in any domain. Some educational tasks or topics are learnable and transferrable more effectively utilizing certain gaming types then others. Furthermore, content has to be divided into different difficulty levels and tasks for distinct audiences. Our game-based learning concept is not limited to a certain audience. For example, undergraduates can learn with gaming content of lower difficulty level and other topics then students in higher education. Game creation is a gaming and learning challenge by itself – so lecturers can choose suitable content out of the gaming pool and add own missing material or further web resources, where necessary. We identified the following domains so far as most suitable to embed for further evaluation purposes: Geography, architecture, history, events, persons, medicine and health. Overall, every domain seems to be suitable for our social educational gaming approach, if learning aims can be fulfilled while creating, answering and querying factual knowledge and predefined learning tasks (especially recall and rearranging of factual content). Popular examples are multiple-choice questions, text-text assignment, image-text assignment or ordering questions. These question types have the advantage, that they are also available as learning standards in Learning Management Systems. They can be easily converted into IMS/QTI after in-game creation. Embedding multimedia like zoom parts out of images or video/audio sequences is also possible. Next to knowledge unit gaming types, we are implementing and researching location-based gaming types relying on geocoded information and correct geographically placement. Here, we can visualize information in a very handsome and effective manner using Yahoo! Maps and their Flash/Flex API to interact on map-based interfaces.

E. Further techniques and libraries

To perform backend and frontend tasks, there are some available and proofed libraries merged up and extended. We are developing qKAI as a Java web application with Servlets and Java Server Pages deployed in a Tomcat Servlet Container [45]. The frontend is next to AJAX designed using Adobe Flex and Flash components for highly interactive tasks with extended design issues (e.g., learning game sequences). Most of the qKAI service and component offer is reusable in nearly any information and knowledge transfer scenario. Additionally some gaming services and components allow specialized functionality for social educational gaming.

F. Project management and build automation

qKAI is using the build and management tool Apache Maven [46] with support of Subversion to automate working tasks. qKAI offers fast start up and good documentation facility of all working processes to reuse them as much as possible this way. Interface classes can be used without knowing about the classes' Weaving. In addition, the standard for (JUnit) tests has been adopted with deploying Maven. The programmers get hints about code snippets where better handling of the framework and its language is suggested. Maven dissolves all dependencies among packages and downloads them automatically in the right version. Developers can use their IDE of choice. Adjusting qKAI to e.g., Netbeans or Eclipse is done through one

Maven command. The server and the database qKAI are currently deployed on, can be changed very easy, too.

VII. Further qKAI use cases and application scenarios

qKAI services enable embedding, querying and enriching distributed web sources for any kind of higher-level application that likes to integrate a broad, structured knowledge base with interoperation ability based on a suitable tools and services collection. We are focusing on process plans suitable for learning - realized by composite services. In the following, we give a few precise examples for use cases that are currently under development using a first prototype of the qKAI framework.

qKAI will publish services in a service broker way for reuse in other web applications like Learning or Content Management Systems.

A qKAI user for instance places a question like "*Which authors are influenced by Stanislav Lem?*" using the SPARQLizer, gets five automatically generated Multiple-Choice answers presented with relational author images from Flickr [47] and he has to choose the right ones out of it. Currently we are developing question-answer-learning-game-types generated out of well suitable Open Data in RDF format. We see two main modes as most interesting for creating first learning game scenarios out of Open Data: Players assign questions to (semi-)automatically extracted information units. Players create questions and get automated answers, which are transformed, into playable answers.

A. qMATCH

This is a prototype of an image-term assignment gaming type. First, the user enters a term he likes to get images about. Then qMATCH presents randomized terms and images out of Flickr and the player has to assign the right term to the right image via Drag & Drop assignment. Here we need a service called wrong-answerizer to assign wrong, but not stupid answers. Wrong-answerizer is deployed in further gaming types. qMATCH is useful to enhance e.g.,, language skills, geographically, architectural or historical knowledge. A conceptual gaming sequence is visualized in Figure 5. If we use term-term assignment, a lot of vocabulary out of various domains can be assessed: assigning English to German translations, assigning buildings to right historical epochs or assigning cities to the right countries.

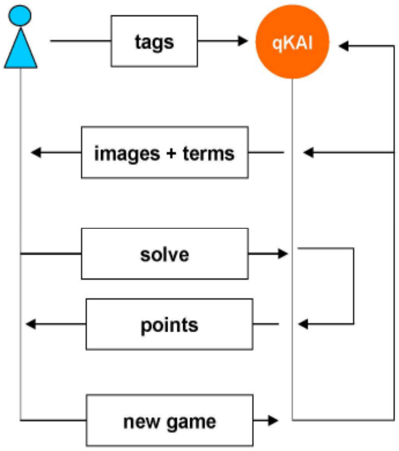


Figure 5. qMATCH gaming sequence

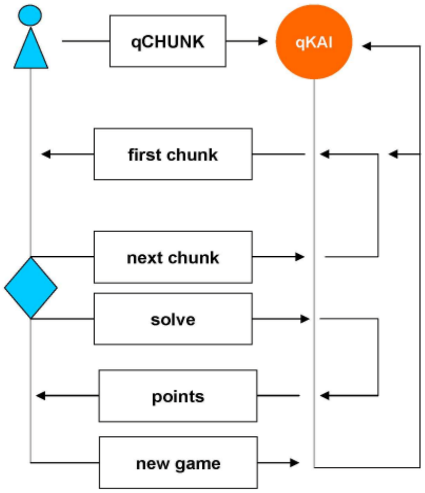


Figure 6. qCHUNK gaming sequence

B. qCHUNK

This is a prototype for a text chunk guessing game based on e.g. Wikipedia articles. qCHUNK presents small textual chunks and the player has to guess the quested term with as less chunks as possible. A conceptual gaming sequence is visualized in Figure 6. Multimedia chunks like zoom parts out of images are conceivable too. The chunks are extracted deploying the SentenceDetector of OpenNLP [18]. The guessable term is exchanged with a placeholder like "?" and is displayed to the user. The user gets 20 seconds to solve or to switch to the next chunk related to the guessable term. qCHUNK is also suitable to integrate multimedia e.g.,, images, sounds or videos.

Example:

- Chunk: ?? is the capital of Lower Saxony founded in 1942.
- Answer: Hanover.
- Next chunk: ?? is located at the Leine.
- Next chunk: image of Hanover city.

Often chunks imply indirect questions like "*Is ?? the capital of Lower Saxony*". The user will get the ability to create a question out of the content that is displayed. While storing and processing hypermedia, it is mandatory that chunks do not lose their origin context and provenance.

C. qMAP

With qMAP a map-based geocoding frontend is under development. Questions, answers and their combination (knowledge units) will be placed and enriched with geocodes at the qMAP. qMAP acts as a kind of gaming board for placing, exploring and editing stored question-answer-pairs. The qMAP will interact with SPARQLizer (see Section D) and represents submitted questions and related answers.

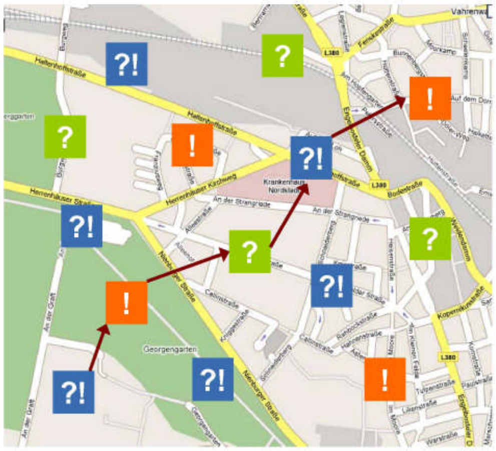


Figure 7. qMAP concept

qMAP offers placement of locations, events, buildings, photos or persons. We provide interaction like filtering, searching, editing and adding knowledge. OpenStreetMap [48] or Yahoo! Maps [49] are good alternatives to Google Maps [50]. Map symbols are connected with different gaming interactions and information units.

D. *SPARQLizer*

With the SPARQLizer a visual query interface is under design, that allows intuitive question to query plan transformation on distributed RDF stores deploying SPARQL endpoints and dynamic graph interlinking. User-generated SPARQL queries are stored as graphs enlarging the qKAI knowledge base and ready to query against in further query plans.

E. *Annotating and qualifying services*

Joker option services for annotating, rating and qualifying are currently under development belonging to the interaction manager of the mediation layer. qKAI jokers allow game-based functionality to add additional sources and to qualify metainformation by rating and ranking input to the qKAI knowledge base. Playing the "*Know-it-all-Joker*" bounds the player to add a source (or information) that proves contrary statements. The "*Nonsense-Joker*" marks an information unit as semantically wrong or inconsistent and defers it to review mode by other qKAI users. The "*Hint-Joker*" allows looking up related sources or other users' answers as solution suggestion. The "*Explorer-Joker*" allows exploring the right answer on the web outside of qKAI during a predefined time. The "*History-Joker*" enables lookups in played answers, ratings of other users by logged interaction and transaction protocols. Statistical protocol analysis is suitable to infer further metainformation.

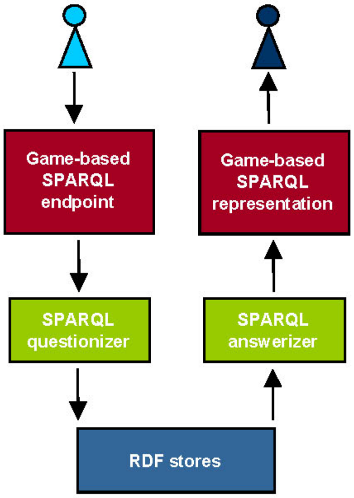


Figure 8. SPARQLizer concept

F. *Social knowledge profiles*

Personal profiles and self-reputation are a popular, ongoing web trend. There are useful Single-Sign-On solutions like OpenID [51] or FOAF [52] files to avoid redundant profile storage in different web communities. Knowledge and learning related properties are enlightened less. qFOAF aims at building personalized, transparent knowledge profiles to connect users additionally thematically and semantically. qFOAF profiles are visible to other users and list a statistical overview of the own knowledge activity. Categories applied contain domain interest and expert level with detailed scores and ranking. This profile serves allied and alienated players as a hint for further activity as known in common games. qFOAF builds a qKAI resource in RDF at the beginning of a game as extended FOAF file with unique URI for every user. It connects the user with topics, knowledge units or other players systematically while gaming. The qFOAF file can be enriched (semi)automated with given information by the user. Existing FOAF files, geocodes or interests, can be included while gaming and interacting with game points and content. Examples for gaming content are created questions, answers, knowledge units, ratings, resources, domains, locations or friends.

G. *Global point and level system*

A global point system is provided to document learning progress, personal interests and to implement incentive and reputation ability. Every kind of interaction is rewarded with qPoints according to its grade of interoperation.

Incentive for user participation is implemented as globally rewarding system of any interaction (qPOINT, qRANK). "*Knowledge is power*" is the simple conceptual slogan on top of qKAI gaming. The more users interact and adapt knowledge, the more they will be rewarded. Adapting knowledge is possible while solving learning tasks on your own or in alliances. Single player and alliances can challenge with each other offering knowledge battles of certain topics. A look into qFOAF user or alliance profiles allows to estimate the own chance to win a challenge or battle. We will enable to steal knowledge from others. Knowledge will become conquerable this way. Alliances and single player will be able to own others knowledge by solving game tasks. An underlying point based rewarding system can be converted into avatar items and further awards. In the future, gaming with qualified content might bring gaming points into reality by allocating them with test examinations to offer further incentive to students. The global point and level system documents learning progress and personal knowledge. qRANK services allow game-based rating and ranking of resources by levels and in-game joker options. After users have played jokers, ranking, annotation and even semantic correction becomes possible.

VIII. Conclusion and future work

We introduced the qKAI application framework for utilizing arbitrary Open Data sources and services in a standardized, RESTful manner aiming at highly interactive scenarios in information and knowledge transfer. To keep qKAI easy extensible with reusable, autonomous service design, we added next to a slim, RESTful SOA a 4-layered mediator-wrapper-schema to the system specification. qKAI combines Web 2.0, Semantic Web and SOA paradigms to apply and enable the Internet of Services for higher level, user-oriented applications with open access.

We are implementing first educational gaming prototypes and we are gaining at further features to utilize current web trends like social interaction for learning purpose and sustainable, knowledge-related interoperation on Open Content. Current focus lies on implementing use cases like the SPARQLizer and qMAP based on the qKAI data layer. SPARQLizer needs a dynamic, adaptive GUI, template and ontology structure to support users in (semi)automated question and answering generation out of SPARQL requests. Chunking information into capable knowledge units is work in progress. The specification of qKAI information units is done and exemplary implemented based on live Wikipedia content. Requirements are derived to utilize Open Content for social educational gaming and standard tasks in knowledge engineering. Atomic services and frontend components allow rearrangement and are suitable for any other purpose in information and knowledge transfer – not only for game-based learning. Services are adaptable to several domains.

Reusing and composing is a precept at all levels in qKAI, the challenge is to merge and to expand: Resources, ontologies, web services or available frameworks. We are aiming at standardized, machine- and human readable staging of Open Content with lightweight interoperation to develop incentive, web-based knowledge transfer and learning scenarios. Therefore, we deploy Linked Data, Rich Clients and REST. The qKAI application framework serves as conceptual basis and system specification for further work exemplary highlighted in this contribution. We want to offer learning scenarios based on user-oriented web services with lightweight interaction grounded on Open Content. Therefore, we have to implement standard tasks of knowledge engineering for extended interaction as a generic application framework. qKAI combines and respectively extends available Java APIs for subtasks to a scalable, reusable and unifying software concept.

Overall, our research focus lies on three main aspects:

- Provide standard tasks of knowledge engineering (acquisition, formalization, representation, visualization).
- Determine and enhance quality of content (analyzes and enrichment of metainformation. User's opinion and knowledge serves to annotate, rate and rank content).
- Tackle extended interaction and incentive for user's attendance while interoperating with Open Content.

We implemented an aware compact and hybrid data layer that serves as basis for further knowledge-processing applications and as interface to semantic search engines and RDF stores. The up to now missing possibilities of the therefore designed query language SPARQL are substituted by the transformation into a relational database schema. Resulting models are flexible and reusable, because any relational data structure can be mapped onto. The chosen approach of parallel data fetching and processing was not that easy to implement concerning thread safe programming, but it is very profitable regarding sustainable deployment - independent of external, at this time still unsecure circumstances:

- Time intensive requests will not block following queries in qKAI.
- Only this way data can be loaded additionally with adequate response times, if a request is not answerable through the pre-buffered data store.

Our research showed that up to now the power of SPARQL is not yet applicable to the efforts of the qKAI data layer. Even if suggested extensions like OpenLink [17] are integrated into the final specification, it is not obvious up to now, whether SPARQL alone will be properly suited for large, distributed data requests soon. Currently we noticed an ongoing trend to deploy relational database concepts because of their majority and performance – like nearly all SPARQL endpoint and RDF repository implementations do [53]. Future work in qKAI is to implement available algorithms and APIs to crawl distributed resources and to amplify the optimization of the qKAI data layer. Some MySQL optimizations like the Sphinx full text indexer [54] are implemented yet to allow faster search. For geographically search space, a GIS extension is ready to use for periphery searches.

All over, the interface to Open Data resources over SPARQL endpoints offers new ways to combine, enrich and explore the data web as an open, semantically connected graph. Data sources are presented in a standardized, machine interpretable and extensible manner deploying RDF repositories. They offer many advantages like open Linked Data ability and annotation compared to traditional relational databases under closed world assumption. Further SPARQL functionality is under development and will be part of its specification soon. We decided to mix up proven and new concepts to get a nowadays practically data layer implementation while buffering preselected data dumps in a relational database and acquiring Open Content out of RDF repositories using SPARQL. Additionally we developed with POI setting for applications and users a practically solution to update pre buffered resources partially on demand when they are requested.

ACKNOWLEDGMENT

The BMBF (Bundesministerium für Bildung und Forschung) co-funded this work under the HELCA (Hannover E-Learning Campus) project. We thank especially Mark Kubacki and Jan Hein for their support in implementing and advancing the qKAI application framework.

REFERENCES

[1] M. Steinberg, J. Brehm, “Towards utilizing Open Data for interactive knowledge transfer”. Proc. DigitalWorld 2009, International Conference on Mobile, Hybrid, and On-line Learning (IARIA’s eLmL), IEEE Press, 2009, pp.61-66, doi:10.1109/ eLmL.2009.13.

[2] T. Berners-Lee, “Web 2.0 and Semantic Web”, http://www.w3.org/2006/Talks/1108-swui-tbl/#(1), W3C, visited 10.04.2009.

[3] M. Domenig, “Rich Internet Applications und AJAX”, Entwickler-Magazin, http://www.canoo.com/news/entwickler.pdf, 2006, visited 15.05.2009.

[4] DBpedia, http://dbpedia.org/About, visited 20.05.2009.

[5] Freebase, http://www.freebase.com, visited 25.05.2009.

[6] Twine, http://www.twine.com/, visited 27.05.2009.

[7] H. Mandl, G. Reinmann-Rothmeier, “Unterrichten und Lernumgebungen gestalten”, Forschungsbericht Nr. 60, Instituts für Pädagogische Psychologie und Empirische Pädagogik, München, Ludwig-Maximilians-Universität, 1995.

[8] IMS/QTI, http://www.imsglobal.org/question/, IMS Global Learning Consortium, Inc., visited 25.05.2009.

[9] C. Schroth, T. Janner, “Web 2.0 and SOA: Converging Concepts Enabling the Internet of Services”, IT Professional, Volume 9, Issue 3, 2007, pp 36 – 41.

[10] G. Wiederhold, “Mediators in the Architecture of Future Information Systems, IEEE Computer, Journal, 25(3), 38-49, 1992.

[11] Open Knowledge Foundation, The Open Knowledge Definition, http://opendefinition.org/, visited 25.05.2009.

[12] RDF, http://www.w3.org/RDF/, W3C, last update: 2008.

[13] SPARQL, http://www.w3.org/TR/rdf-sparql-query/, W3C, visited 25.05.2009.

[14] J. H. Dieter Fensel, H. Lieberman, and W. Wahlster, “Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential”, The Mit Press, Massachusetts, 2005.

[15] C. Bizer, R. Cyganiak, “D2R server: Publishing Relational Databases on the Semantic Web”, http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/, visited 27.05.2009.

[16] Aperture, http://aperture.sourceforge.net/, Aduna, DFKI, visited 27.05.2009.

[17] Openlink Virtuoso, http://virtuoso.openlinksw.com/, visited 27.05.2009.

[18] OpenNLP, http://opennlp.sourceforge.net/, visited 27.05.2009.

[19] F. Naumann, “Quality-Driven Query Answering for Integrated Information Systems”, Lecture Notes in Computer Science Vol. 2261, Springer, 2002.

[20] C. Becker, C. Bizer, “DBpedia Mobile: A Location Enabled Linked Data Browser”, 17th International World Wide Web Conference WWW2008, China, 2008.

[21] T. Heath, E. Motta, “Revyu.com: a Reviewing and Rating Site for the Web of Data”, Proc. ISWC 2007, International Semantic Web Conference, Lecture Notes in Computer Science 4825 Springer 2007, pp. 895-902.

[22] L. Ahn, S. Ginosar, M. Kedia, R. Liu, and Manuel Blum, “Improving Accessibility of the Web with a Computer Game”, International conference for human-computer interaction, CHI 2006, Canada.

[23] GuessTheGoogle, http://grant.robinson.name/projects/guess-the-google/, visited 27.05.2009.

[24] Scoyo, www.scoyo.de, visited 27.05.2009.

[25] Playfish, www.playfish.com, Megazebra, www.megazebra.com, Zynga, www.zynga.com, visited 27.05.2009.

[26] Facebook, www.facebook.com, MySpace, www.myspace.com, visited 27.05.2009.

[27] ILIAS, http://www.ilias.de, visited 27.05.2009.

[28] J. Brehm, M. Steinberg, "Eine Arbeitsumgebung für elektronisch unterstützte interaktive Lehre", 36. GI-Jahrestagung 2006 - Informatik für Menschen, Workshop Pervasive University.

[29] C. Bizer, T. Heath, T. Berners-Lee, “Linked Data: Principles and State of the Art”, 17th International World Wide Web Conference, WWW2008, China, 2008.

[30] R. Fielding, “Architectural Styles and the Design of Network-based Software Architectures”, Dissertation, Irvine, 2000.

[31] E. Wilde, R. J. Glushko, “Document design matters”, Commun. ACM 51, 10, 2008, pp. 43-49, doi: http://doi.acm.org/10.1145/1400181.1400195.

[32] Adobe Flash/Flex, http://www.adobe.com/products/flex/, visited 27.05.2009.

[33] Knol, http://knol.google.com/k, visited 27.05.2009.

[34] C. Kiefer, Quizzer, http://www.d-elina.de/2008/index.php?option=com_content&task=view&id=97&Itemid=64, last update: 2008.

[35] Prototype, http://www.prototypejs.org/, visited 27.05.2009.

[36] Dojo, http://dojotoolkit.org/, visited 27.05.2009.

[37] YUI, http://developer.yahoo.com/yui, visited 27.05.2009.

[38] OpenRDF Sesame, http://www.openrdf.org/, visited 27.05.2009.

[39] Semantic Web Application Framework, http://doi.ieeecomputersociety.org/10.1109/MS.2007.126, visited 27.05.2009.

[40] Swoogle, http://swoogle.umbc.edu/, visited 27.05.2009.

[41] Sindice, http://www.sindice.com, visited 27.05.2009.

[42] Watson, http://watson.kmi.open.ac.uk/WatsonWUI/, visited 27.05.2009.

[43] JPA, EclipseLink, http://www.eclipse.org/proposals/eclipselink/, visited: 13.05.09.

[44] OpenScreen Project, http://www.openscreenproject.org/, visited 27.05.2009.

[45] Apache Tomcat, http://tomcat.apache.org/, visited 27.05.2009.

[46] M. Loukides, „Maven - The Definitive Guide“, O´Reilly Media, Inc., Sebastopol, 2008.

[47] Flickr, http://www.flickr.com/, visited 27.05.2009.

[48] OpenStreetMap, www.openstreetmap.de/, visited 27.05.2009.

[49] Yahoo! Maps, http://maps.yahoo.com/, visited 27.05.2009.

[50] Google Maps, www.maps.google.com, visited 27.05.2009.

[51] OpenID, http://openid.net/, visited 10.04.2009.

[52] FOAF, http://xmlns.com/foaf/spec/#term_name, visited 27.05.2009.

[53] M. Völkel, SparQL Implementations, ESW Wiki, 2008, http://esw.w3.org/topic/SparqlImplementations, visited 27.05.2009.

[54] J. Kumar, “Benchmarking results of mysql, lucene and sphinx”, 2006, http://jayant7k.blogspot.com/2006/06/benchmarking-results-of-mysql-lucene.htm, visited 27.05.2009.

Web Services Solutions for Hydrologic Data Access and Cross-Domain Interoperability

Bora Beran
Microsoft Research
Redmond, Washington, USA
borabe@microsoft.com

David Valentine, Ilya Zaslavsky
San Diego Supercomputer Center
University of California, San Diego
La Jolla, California, USA
valentin@sdsc.edu, zaslavsk@sdsc.edu

Simon Jonathan David Cox
Commonwealth Scientific and Industrial Research Organization (CSIRO)
Bentley, Australia
simon.cox@csiro.au

John McGee
Renaissance Computing Institute
Chapel Hill, North Carolina, USA
mcgee@renci.org

Abstract— **Agencies such as US Geological Survey (USGS), Environmental Protection Agency (EPA), National Oceanographic and Atmospheric Administration (NOAA) offer considerable amount of data on climate, hydrometry and water quality in the United States spanning from 1860s to the current day. While accessible through a web browser, data from these sources typically cannot be directly ingested by modeling or analysis tools without human intervention. Different input/output formats, syntax and terminology, and different analysis scenarios the systems were designed to support, make data discovery and retrieval a major time sink. This paper examines the web services developed as a part of Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information System (HIS) project as a means to standardize access to hydrologic data repositories, facilitate data discovery and enable direct machine-to-machine communication, and the efforts in larger scale to create a standard which is more flexible and generic yet capable of capturing the domain semantics such that interoperability with other scientific domains can be achieved losslessly.**

Keywords- Webservices; interoperability; international standards; geosciences; hydrology

I. INTRODUCTION

The world is facing major challenges associated with the environment particularly around climate change and water scarcity. Changing temperature patterns cause hydrologic cycle to become less predictable while pollution and increasing demand for water due to population growth are pushing the limits of sustainability. Coping with these issues require working across disciplines with data of varying temporal and spatial scales. For instance while flood warning systems rely on near real-time data, understanding climate change and drought patterns or making engineering decisions about structures such as dams or levees require historical data which can be in-situ point observations as well as remote sensing imagery.

In the US, Environmental Protection Agency (EPA), US Geological Survey (USGS) and National Oceanographic and Atmospheric Administration (NOAA) are the primary sources of water quality, quantity and climate data. While there are overlaps in data offerings NOAA is the main source of meteorological data, USGS stands out with its extensive water quantity (surface/subsurface) data whereas EPA focuses on environmental quality. Heterogeneity is a major issue. USGS data is available, via the National Water Information System (NWIS) in different formats including delimited text, HTML tables and USGS' own HydroML markup language. EPA is moving from delimited text to XML-based WQX (Water Quality eXchange) format. In addition to different encodings, there is no common vocabulary either. Lack of standards for hydrologic data exchange is a major problem a solution to which would eliminate the need for human involvement in data retrieval thus not only saves valuable research time but also makes it possible to implement automated workflows. This has been the main motivation behind the water data services part of the Consortium of Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic Information Systems (HIS) project [1]. The project's experience in developing web for standardized access to hydrologic data sources in the United States demonstrates the challenges associated with establishing community semantics of hydrologic data exchange, formalizing the main notions of hydrologic observations, and evolution towards compliance with general data exchange protocols for cross-domain interoperability. However international aspects should also be taken into account as 145 nations have territory in the 263 trans-boundary river basins in the world and approximately one third of these basins are shared by more than two countries [2].

II. DATA COVERAGE

According to surveys, in the United States 60.8% of hydrologists in academia consider NWIS stream flow data necessary for their research [3]. NWIS is followed by NOAA's National Climatic Data Center (NCDC) precipitation data (35.1%). NCDC pan evaporation, NWIS groundwater levels, Environmental Protection Agency (EPA) Storage and Retrieval System (STORET) water quality, National Land Cover Dataset, National Elevation Dataset, State Soil Geographic (STATSGO) & Soil Survey Geographic (SSURGO) datasets, National Hydrography Dataset and remote sensing data (e.g. LANDSAT) are other datasets in the list. The CUAHSI HIS focused its attention first on the NWIS and EPA STORET as hydrologists' top two preferences with nationwide coverage and freely available data. Development of web service wrappers for hydrologic repositories at these two agencies were followed by services for Moderate Resolution Imaging Spectroradiometer (MODIS), North American Mesoscale Model (NAM) and Daily Meteorological Summaries (DAYMET) data which present gridded time series for common weather and climate variables. In addition, the hydrologic data publication workflow developed by the project, allowed other research groups, from state and local governments, academia and non-profit environmental organizations, to make their hydrologic measurements accessible through the system. The data were loaded or streamed into the CUAHSI Observations Data Model [4], exposed via the common set of web services, and registered to the CUAHSI HIS Central portal; currently over 50 community-generated data sources are published in this way.

III. HETEROGENEITY PROBLEM

Syntactic, semantic and information system disparities between web-accessible hydrologic repositories complicate their integration. To a large extent, the heterogeneities derive from the differences in the use cases envisioned in each of the agency systems, data collection and management practices, information models and internal data structures. In most cases, these characteristics are not explicitly expressed or available for review. Hence, the interoperability solutions are necessarily limited, as we attempt to capture the core semantics of hydrologic data discovery and retrieval common across different systems, and define system-specific extensions that reflect the specific intent and use cases of each agency system. Information system heterogeneity is a result of different interfaces and/or communication protocols.

Semantic heterogeneity occurs when there is no prior agreement about the meaning, interpretation or intended use of the same or related data [5]. For example equivalent measurement units can appear to be different due to several reasons such as use of different abbreviations and notations, or even typos. Table 1 gives a few examples of these differences (and errors). In the course of Water Markup Language (WaterML) 1.0 development approximately 900 units used by target repositories were reduced down to 302 common units by fixing these errors and making use of equivalences. Two mechanisms have been used within the CUAHSI HIS project to tame semantic heterogeneity. Controlled vocabularies for commonly used fields, such as units, spatial reference systems, sample medium, censor codes, etc., are managed by an online Master Controlled Vocabulary Registry available at http://his.cuahsi.org/mastercvreg/cv11.aspx and published as SOAP services, to enable vocabulary validation at the client applications. For such fields where the use of controlled vocabulary is problematic (e.g. measured parameter names), an ontology-based system is developed that lets data managers associate parameter names in their datasets with concepts in a hydrologic ontology, thus enabling semantics-based search across different repositories regardless of variable naming preferences of individual systems [6].

TABLE 1.SEMANTIC HETEROGENEITY IN MEASUREMENT UNITS

Source 1	Source 2	Note
acre feet	acre-feet	punctuation difference
micrograms per kilogram	micrograms per kilgram	spelling error
FTU	NTU	equivalent
mho	Siemens	equivalent
ppm	mg/kg	equivalent

Syntactic heterogeneity is the presence of different representations or encodings of data. Date/time formats can be given as an example where common differences are; local time vs. UTC, 12 hour clock vs. 24 hour clock and Gregorian date vs. Julian day which is common in Ameriflux data.

IV. CUAHSI WEB SERVICES

The goal of CUAHSI web services is to reconcile the aforementioned differences to the extent possible and return uniform documents regardless of the repository of origin. Hence CUAHSI HIS web services have been named WaterOneFlow; emphasizing the idea of a seamless interface through which researchers can gain access to hydrologic data from multiple heterogeneous data sources.

TABLE 2. WATERONEFLOW WEB SERVICE METHODS

Methods	**Description**
GetSiteInfo, GetSiteInfoObject	Given a site number, this method returns the site's metadata. Send the site code in this format: 'NetworkName:SiteCode'
GetSites, GetSitesObject	Given an array of site numbers, this method returns the site metadata for each one. Send the array of site codes in this format: 'NetworkName:SiteCode'
GetValues, GetValuesObject	Given a site number, a variable, a start date, and an end date, this method returns a time series. Pass in the sitecode and variable in this format: 'NetworkName:SiteCode' and 'NetworkName:Variable'
GetVariableInfo, GetVariableInfoObject	Given a variable code, this method returns the variable's name. Pass in the variable in this format: 'NetworkName:Variable'

WaterOneFlow follows certain rules to ensure uniformity of both input and output communication with the services. To this end web services were designed to provide output in a standard format; namely CUAHSI WaterML as part of the CUAHSI HIS project. The main purpose of WaterML has been to encode the semantics of discovery and retrieval of hydrologic time series, as commonly used by research hydrologists. This domain semantics has been derived from the CUAHSI Observations Data Model as well as from the organization, data structures and metadata exposed by several common online repositories of water quantity and water quality data. WaterML has been developed as a set of core constructs (site, variable, timeseries, etc) reflecting a common usage scenario where time series are discovered and retrieved by navigating to sites of interest and then examining parameters measured at these sites and their periods of record. As a result, WaterML offered an attractively simple formal encoding of time series exchange, which was implemented in WaterOneFlow services and field tested within a distributed system of hydrologic observatory test beds. WaterOneFlow services offer four major functions and their variants. (See Table 2) Object suffix (e.g. GetValuesObject) indicates that method returns a WaterML created by deserializing the response into an object, rather than WaterML being returned as a String. Different options are provided for users of varying levels of programming experience and not necessarily the same preferences.

```
# Data for the following site(s) are contained in
this file USGS 06090800 Missouri River at Fort
Benton MT
#
# -----------------------------------------------
# Data provided for site 06090800
#    DD parameter   Description
#    02   00060     Discharge, cubic feet per
second
#
agency_cd site_no   datetime  02_00060  02_00060_cd
5s        15s       16d       14n       10s
USGS      06090800  2009-09-06 04:00    5750      P
USGS      06090800  2009-09-06 04:15    5780      P
USGS      06090800  2009-09-06 04:30    5780      P
USGS      06090800  2009-09-06 04:45    5780      P
```

Figure 1. Sample USGS NWIS response to a data request

Figure 1 shows the output of a USGS NWIS inquiry for discharge measurements at site number 0609800 between 4:00 AM and 4:45 AM on September 6th, 2009. Figure 2 shows the response of WaterOneFlow GetValues service to the same data request. It can easily be seen that Figure 2 contains significant amount of metadata lacking in the original USGS response. Coordinates of measurement location, type of measurement (instantaneous, average, minimum, maximum, incremental etc.) and time zone are some of the additional content very important for correctly interpreting the data. This is because WaterOneFlow services are not just proxies that transform the data but are supported by a local metadata catalog or they retrieve the additional information by making several different inquiries to underlying data repositories.

WaterOneFlow services for national datasets and hydrologic observatory test-beds are operational and can be accessed at http://river.sdsc.edu/wiki/CUAHSI%20WebServices.ashx and http://www.watersnet.org/wtbs/, respectively.

There are two main deployment scenarios for WaterOneFlow services. If data is contained in CUAHSI HIS' Observations Data Model (ODM), the deployment is fairly straightforward. A different scenario is implemented when the data are housed in a remote repository such as a federal agency database accessible via a Web interface. In such cases, WaterOneFlow services can be screen scraper services aka web service wrappers. This is an error-prone approach as the services are sensitive to slight alterations of the remote web site. This bottleneck is removed as water data collection agencies develop web service interfaces to their repositories. Data repositories such as NCDC Automatic Surface Observing System (ASOS) and USGS NWIS have implemented WaterOneFlow webservices on their servers, eliminating the need for screen scraping. More repositories are expected to follow.

```
<timeSeries>
  <sourceInfo xsi:type="SiteInfoType">
    <siteName>Missouri River at Fort Benton MT</siteName>
    <siteCode network="NWIS">06090800</siteCode>
    <timeZoneInfo>
      <defaultTimeZone ZoneAbbreviation="MST" ZoneOffset="-07:00" />
      <daylightSavingsTimeZone ZoneAbbreviation="MDT" ZoneOffset="-06:00" />
    </timeZoneInfo>
    <geoLocation>
      <geogLocation xsi:type="LatLonPointType" srs="EPSG:4269">
        <latitude>47.81746979</latitude>
        <longitude>-110.6671586</longitude>
      </geogLocation>
    </geoLocation>
  </sourceInfo>
  <variable>
    <variableCode vocabulary="NWIS">00060</variableCode>
    <variableName>Discharge</variableName>
    <variableDescription>Discharge, cubic feet per second</variableDescription>
    <dataType>Instantaneous</dataType>
    <units unitsAbbreviation="cfs">cubic feet per second</units>
    <NoDataValue>-999999</NoDataValue>
    <timeSupport isRegular="true">
      <unit>
        <UnitName>minute</UnitName>
        <UnitType>Time</UnitType>
        <UnitAbbreviation>min</UnitAbbreviation>
      </unit>
      <timeInterval>15</timeInterval>
    </timeSupport>
  </variable>
  <values count="4">
    <value dateTime="2009-09-06T11:00:00-7:00" qualifiers="P">5750</value>
    <value dateTime="2009-09-06T11:15:00-7:00" qualifiers="P">5780</value>
    <value dateTime="2009-09-06T11:30:00-7:00" qualifiers="P">5780</value>
    <value dateTime="2009-09-06T11:45:00-7:00" qualifiers="P">5780</value>
  </values>
</timeSeries>
```

Figure 2. Excerpt from WaterOneFlow GetValues response

V. APPLICATIONS OF WATERONEFLOW SERVICES

WaterOneFlow services have been leveraged by several applications with purposes ranging from data discovery to hydrologic & water quality modeling. Macros and toolbars developed for Microsoft Excel, Matlab and ArcGIS allow importing data directly into these applications [7]. Web-based applications such as Data Access System for Hydrology (DASH) [8] and Hydroseek [6] facilitate data discovery and retrieval by providing unified map-based interfaces over multiple repositories (Figure 3).

Applications in water resources modeling make use of Open Modeling Interface (OpenMI). OpenMI defines an interface that allows time-dependent models to exchange data at run-time. Goodall et al. developed models to calculate watershed storage and water quality [9].

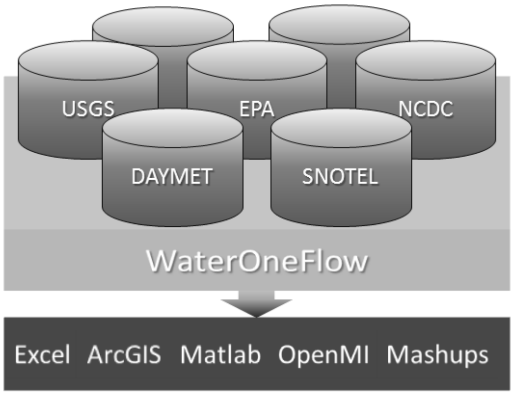


Figure 3. WaterOneFlow as a bridge to analysis and data discovery tools

Storage model is an application of conservation of mass principle and uses precipitation, streamflow (inflow-outflow) and evapotranspiration data from USGS NWIS, DAYMET and Ameriflux repositories respectively. Water quality calculations leverage USGS's SPAtially Referenced Regressions On Watershed attributes (SPARROW) model. SPARROW model performs the regression on total nitrogen loadings derived from observations of organic nitrogen, inorganic nitrogen, and flow. In this particular implementation USGS stations measuring streamflow are used along with nearby EPA stations with nitrogen concentration measurements to obtain the necessary data for the model. Once the observation results are retrieved from USGS NWIS and EPA STORET they are aligned in space and time and used as model input.

VI. INTEROPERABILITY, THE BIG PICTURE

WaterML and WaterOneFlow services have established an initial level of interoperability across hydrologic data repositories that reflected the semantics of water data discovery and retrieval common in hydrologic research. Their implementation in the context of an operational distributed system of the CUAHSI HIS project providing web service access to data measured at over 1.75 million sites in the US, allows the project team to further specify use cases and scenarios, and additional requirements for a hydrologic data exchange format. To address interoperability challenges beyond the hydrology domain, and accommodate additional usage scenarios, the approach has to be extended and harmonized with emerging standards in other domains. Several such standards are being developed under the aegis of the Open Geospatial Consortium (OGC).

A hydrology domain working group has recently been convened within OGC, to focus on formulation of interoperability requirements and scenarios in hydrology, and coordinate the development of a common exchange protocol, referred to as WaterML 2.0 operating alongside a meteorology working group under the umbrella of the OGC's Earth System Science domain working group. As part of this process WaterML is being harmonized with OGC standards for sensor/geographic data exchange to become interoperable with similar applications from different domains.

A. *Sensor Web Enablement*

Open Geospatial Consortium (OGC) provides a framework that specifies standard interfaces and encodings to facilitate exchange of geographical information. OGC's Sensor Web Enablement (SWE) initiative focuses on integration of sensors and sensor systems [10].
SWE develops standards to enable:

- Discovery of sensor systems and observations
- Determination of a sensor's capabilities
- Retrieval of sensor metadata
- Retrieval of time-series observations and coverages
- Subscription to and publishing of alerts to be issued by sensors based on certain criteria
- Tasking of sensors

The principal SWE service interface (related to the top four bullets) is called Sensor Observation Service (SOS). SOS [11] uses the OGC information standards Observations & Measurements (O&M) [12] and Sensor Model Language (SensorML) [13] for encoding observations data/metadata and sensor metadata respectively. Sensor Alert Service (SAS) and Sensor Planning Service (SPS) [14] define interfaces for subscription and tasking. SOS occupies the *services* tier shown in Figure 4. This may be compared with the IEEE 1451 family of standards which addresses the transducer interface tier.

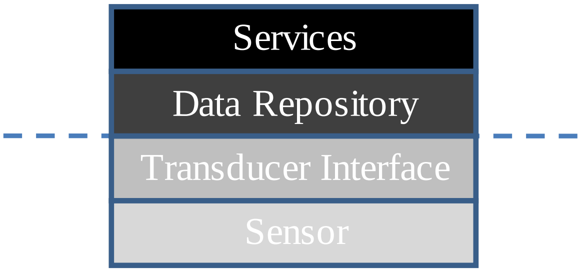


Figure 4. Generalized SWE stack

SOS defines three core and mandatory operations:

- **GetObservation** for retrieving sensor data
- **GetCapabilities** for retrieving information about the data offerings and supported functions (e.g. filters) for the service instance
- **DescribeSensor** for retrieving sensor metadata

A typical sensor data consumption scenario starts with service discovery which involves using one or more OGC Catalog Service (CS-W) [15] instances. CS-W provides an interface to a registry allowing data consumers to discover services by time period of observations, phenomena captured by observations, spatial extent, names and descriptions. Evaluating the suitability of a specific service instance utilizes the GetCapabilities operation. A GetCapabilities response contains detailed information about all of the offerings that are available from a SOS instance, which typically exposes a small constellation of sensors, details of which may be obtained through the DescribeSensor operation. GetCapabilities response also contains information on the filters supported by GetObservation operation. Filters are used to subset observation results based on temporal, spatial, logical or scalar comparison operators [11].

The SOS interface is optimized to deliver sensor-generated observations, where an observation is defined as an act that uses a procedure to determine the value of a property or phenomenon related to a feature of interest. SOS is a generic interface to observation data from any discipline. Observation semantics are provided by the definition of the *feature of interest*, the *observed property*, and the *procedure* used in generating observation results. These must be defined in the context of a particular application domain, maintained separately from the generic interface definition. The procedure may involve a sensor or observer, analytical procedure, simulation or other numerical process [12, 13].

TABLE 3. COMPARISON OF WATERONEFLOW AND SOS METHODS

Sensor Observation Service	**WaterOneFlow**	**Comments**
GetCapabilities	GetSites, GetSiteInfo	Site IDs (known as 'feature of interest' in SWE) are included in the GetCapabilities response. Capabilities are identical in all WaterOneFlow instances. GetCapabilities response contains a list of offerings, analogous to list of time series returned by WaterOneFlow.
DescribeSensor	GetVariableInfo	WaterOneFlow does not provide access to sensor or procedure descriptions. However some sensor properties are provided as part of the description of the observed variable.
GetObservation	GetValues	-
GetFeatureOfInterest	GetSiteInfo	-
DescribeObservationType	GetVariableInfo	-
DescribeFeatureType	-	Since SOS is generic, there is a specific operation to get a description of the subject of the observations. Whereas WaterML 1.0 has observation site as the only feature type.
GetFeatureOfInterestTime	GetSiteInfo	The time(s) that a mobile sensors observes a particular feature
GetResult	-	Lightweight access to values, with no metadata
DescribeResultModel	-	Since SOS/O&M are generic, a variety of result encodings may be used. This operation retrieves an explicit description of the encoding.

An SOS instance may be backed by a variety of data sources, which may be live sensors, but commonly is a data store which caches observation data. Such a cache may itself be updated through other SOS interface(s), but will commonly be updated through a private interface. (Early version SOS prototypes were even based on scraping HTML pages.) SOS merely provides a standardized http-hosted interface and request syntax, essentially a standard façade for less convenient data sources, which make them appear like a 'virtual XML document'.

Even though SOS is a fairly new standard, it is possible to see many implementations in different domains and parts of the world as an indicator of its potential to facilitate cross-domain interoperability. OOSTethys/OceansIE (Marine Science), Open architecture for Smart and Interoperable networks in Risk management based on In-situ Sensors (OSIRIS), Sensor Asia (Landslide warning, Drought monitoring) [16], Water Resources Observation Network (WRON) are examples from the United States, Europe, Asia and Australia respectively. In fact experiences from WRON project in South Esk River Catchment in the north-east of Tasmania contribute to WaterML 2.0 development. The WRON implementation communicates directly with the sensor, in contrast to WaterML 1.0 which was targeted primarily at data repositories.

B. *Water Observations Markup Language*

Water Observation Markup Language (WOML) is an application of OGC's Observations & Measurements (O&M) and Sensor Observation Service (SOS) standards for the hydrology domain. It was developed as a proof-of-concept, to evaluate the ability of the OGC standards to match the scope of WaterML v1.0.

O&M [12] decouples the generic model for an observation (with an 'observed property', 'feature of interest' 'procedure', and 'result') from the domain-specific semantics (e.g. the definition of 'stream' or 'watershed'). The latter must be provided by a separate schema, specialized for the application domain. However, recognizing that spatial sampling strategies are common across the natural sciences, O&M Part 2 [17] provides standard sampling feature types such as 'sampling point', 'sampling curve', 'specimen', which correspond to stations, profiles, transects, wells, sample etc. WOML also differs from WaterML in using controlled vocabularies from external authorities, in preference to local definitions. For example, the O&M XML implementation is a Geography Markup Language (GML) application [18], within which the Unified Code for Units of Measure (UCUM) codes [19] is recommended (when suitable). Hence WOML uses UCUM codes to scale measurement results. GML provides standard patterns for the use of URIs to link to external resources, to enable and encourage the use of pre-existing externally governed vocabularies. In this way both data structures and key aspects of the content are standardized, which leads to improved interoperability. So overall WOML is composed from O&M Parts 1 and 2, plus a lightweight domain model for hydrology (watercourse, storage, catchment), some standard vocabularies of units, sensors, interpolation rules, and state behavior and request metadata provided by SOS.

TABLE 4. SEMANTIC DIFFERENCES OVER SHARED CONCEPTS BETWEEN DATA FORMATS

WaterML 1.0	**NWIS**	**STORET**	**WQX**	**SOS**
Site	Site	Station	Monitoring Location	Feature
Lat-Long	Lat-Long	Lat-Long	Lat-Long	Arbitrary geometry (may be point coordinates)
Variable	Parameter	Characteristic	Characteristic Name	Observed property
Method	Parameter	Method	Method	Procedure
Series	Period of Record	-	Characteristic Summary	Offering

Through WOML work, use cases and experiences from WaterML and WaterOneFlow in turn are contributing design considerations for OGC standards under development. This is also a key benefit to OGC from the formation of the Hydrology working group. For example one of the key challenges in SOS/O&M is encoding time-series. Existing coverage encodings are mostly tailored for imagery, rather than functions with a temporal domain. The WaterML time-series encoding provides a good solution for handling this type of data.

From the point of view of CUAHSI, adopting externally governed standards leads to both benefits and obligations. The benefit of leveraging generic sensor and observation standards is (i) the potential for easier cross-domain data assimilation (important in hydrology, which clearly depends on meteorology, climate science, administrative and engineering information, and geology), (ii) more robust design, based on a broader set of requirements, and (iii) tool re-use. However, there are costs such as (i) dependency on third-party governance and maintenance arrangements for part of the language (ii) complexity due to specialization of a generic component, in contrast to directly designing for a limited use-case (iii) additional conformance constraints that may not be directly relevant to the application domain.

C. *Transition from WaterOneFlow to SOS*

WOML showed that O&M + SOS, customized with hydrology feature-types, property-types (variables or parameters) and sensors can support the functionality equivalent to WaterOneFlow. Table 3 shows how the SOS operations map to WaterOneFlow requests.

One of the principal goals of the OGC Hydrology Working Group is to develop WaterML v2, which will be based on the OGC SWE standards, but will address the detailed requirements identified for the WaterOneFlow services. Looking at Tables 3 and Table 4 it is possible to see that SOS is more generic and atomic, giving it much more flexibility and expressiveness as well as making it easier to parse. However this also makes SOS document structure more complex and less human-readable. While there are many conceptual overlaps at a more abstract level, hypernymy (super-ordinance) and hyponymy (sub-ordinance) are common semantics issues observed between different data sources, both in representations of the data (Table 4) and web service methods (Table 3). A consequence of this is the necessity to deal with much more complex mappings and requirement for wrapper services to often invoke multiple functions of the wrapped system and aggregate the results to be able to respond to a single request.

In order to make the adoption of SOS easier, an open–source SOS implementation is being developed which can be found at http://ogc.codeplex.com/. This work includes class libraries to support SOS and templates to simplify creation of SWE services for the Microsoft .NET environment. Libraries and templates are generic hence can be used outside the CUAHSI HIS framework and with databases other than ODM. However to simplify the migration for existing WaterOneFlow systems, a ready to use out-of-the-box web services/ODM database bundle is also included in the distribution. Operational services can be accessed at http://www.sensordatabus.org/Pages/SOS.aspx.

VII. CONCLUSION

To enable programmatic access to hydrometry/water quality databases in the United States, a set of web services has been developed. Standard web service functions (WaterOneFlow) and a markup language as the medium (CUAHSI WaterML) are used to provide a uniform view over multiple heterogeneous data sources and allow programs and modeling tools directly access and retrieve data from them without need to human intervention. This not only reduces the time spent for data discovery and preparation but also can be used in cases such as scientific work flows. WaterOneFlow services are planned to cover more data sources, offer more functions while WaterML is evolving to become an OGC standard. Web services are an important component in solving the interoperability puzzle by linking the data and applications together. However it is important to have a consensus on a standard otherwise, more time would be spent to make different standards work together. CUAHSI HIS now provides web services to USGS National Water Information System (NWIS), EPA Storage and Retrieval (STORET), Moderate Resolution Imaging Spectroradiometer (MODIS), North American Mesoscale Model (NAM) and Daily Meteorological Summaries (DAYMET) data. Through WaterOneFlow 40 other data sources are available including several international datasets.

To further enhance data interoperability within and beyond the hydrology domain, additional work focuses on harmonizing WaterML development with OGC SWE specifications. While this is a work in progress, WOML and open-source OGC libraries that couple the CUAHSI Observations Data Model with SOS interfaces are important steps towards creation and adoption of a more universal hydrologic data exchange protocol that will be both flexible and generic, at the same time providing intuitive encodings that are compatible with common hydrologic semantics.

ACKNOWLEDGMENT

This work has been partly supported by NSF under award EAR-0622374 (PI: D.R. Maidment), CSIRO and the Australian Bureau of Meteorology under IWIS Theme 1010.2 WIRADA project 1.4 “Water Data Transfer Standards”.

REFERENCES

[1] B. Beran, J. Goodall, D. Valentine, I. Zaslavsky and M. Piasecki, “Standardizing Access to Hydrologic Data Repositories through Web Services”, Proc. International Conference on Advanced Geographic Information Systems & Web Services (GEOWS 2009), IEEE Computer Society, Los Alamitos, CA. Feb. 2009, pp 64-67.

[2] E. Koncagül, D. Donovan, J. Treves-Habar, S. Lee, I. Brugnon, K. Pittman, A. Núñez-Luna, C. Walther and A. Hassan (Eds), “Water: A shared responsibility: The United Nations World Water Development Report 2”, UNESCO, Paris, 2006, pp 126. ISBN: 92-3-104006-5.

[3] D. Maidment (Ed), “CUAHSI HIS Status Report”, September 15, 2005. pp 48-87.

[4] J.S. Horsburgh, D.G. Tarboton, D.R. Maidment and I. Zaslavsky, ”A relational model for environmental and water resources data”, Water Resources Research. vol. 44. doi:10.1029/2007WR006392.

[5] A.P. Sheth and J. Larsen, “Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases”, ACM Computing Surveys: Special Issue on Heterogeneous Databases, 22(3) 183-236.

[6] B. Beran and M. Piasecki, “Engineering new paths to water data”, Computers & Geosciences vol 35, issue 4, Apr. 2009, pp. 753-760.

[7] S.R. Brady, A.K. Sinha and L.C. Gundersen (Eds), “Geoinformatics 2006—Abstracts: U.S. Geological Survey Scientific Investigations Report 2006-5201”, 2006. pp 23.

[8] J. L. Goodall, B.F. Robinson, F.M. Shatnawi and A.M. Castronova, “Linking Hydrologic Models and Data: The OpenMI Approach”, (2007). Eos Trans. AGU, vol. 88(52), Fall Meet. Suppl., Abstract H13H-1682

[9] T. Whitenack, I. Zaslavsky, D. Valentine and D. Djokic, "Data Access System for Hydrology", (2007). Eos Trans. AGU, 88(52), Fall Meet. Suppl., Abstract H13H-1685.

[10] M. Botts, G. Percival, C. Reed and J. Davidson, “OGC Sensor Web Enablement: Overview and High Level Architecture”, Open Geospatial Consortium Inc., OGC 06-052r2, 2006.

[11] A. Na and M. Priest, “Sensor Observation Service”, Open Geospatial Consortium Inc., OGC 06-009r6, 2007.

[12] S. J. D. Cox, “Observations and Measurements, Part 1 - Observation Schema”, Open Geospatial Consortium Inc., OGC 07-022r1, 2007.

[13] M. Botts, and A. Robin, “Sensor Model Language (SensorML)”, Open Geospatial Consortium Inc., OGC 07-000, 2007.

[14] I. Simonis (Ed), “Sensor Planning Service Implementation Specification”, Open Geospatial Consortium Inc., OGC 07-014r3.

[15] D. Nebert, A. Whiteside and P. Vretanos (Eds) “Catalogue Services Specification”, Open Geospatial Consortium Inc., OGC 07-006r1.

[16] K. Honda, D. H. Nguyen and H. Shimamura, “Sensor Asia - Sensor Network and GIS for Environmental Monitoring in Asia”, International Forum on Water Environmental Governance in Asia, pp.102-105, 2007.

[17] S. J. D. Cox, “Observations and Measurements, Part 2 – Sampling Features”, Open Geospatial Consortium Inc., OGC 07-002r3, 2007.

[18] C. Portele (Ed), "OpenGIS Geography Markup Language (GML) Encoding Standard, v3.2.1", ISO 19136:2007, International Organization for Standardization, 2007.

[19] G. Schadow and C.J. McDonald, "The Unified Code for Units of Measure v1.7", Regenstrief Institute, Inc. 2008. <http://aurora.regenstrief.org/~ucum/ucum.html> 2009-09-20

Reliability Issues and Improvements in Remote Monitoring Security Systems

R. Hariprakash
BIHER,Bharath
Univ, Chennai-73
rhp_27@ieee.org

G. Venkat Ramu
University of Madras
Chennai-25, India
ramu_nec@yahoo.com

T. Rajakumari
Polytechnic College
Chennai, India
visa4466@hotmail.com

S. Ananthi
Univ. of Madras
Chenna-25, India
ananthibabu@yahoo.com

K. Padmanabhan
AC Tech, Anna
Univ., Chennai-25,
ck_padmanabhan@rediffmail.com

Abstract – **The paper deals with the methods of security implementation in community areas based on certain high technology sensor system and communication methods. Security systems for home or industry have a lot to perform today in terms of reliability and performance. After examining the various modes for security management, the paper describes data collection and fusion which demand careful processing and discerning of the security problem. Herein, the problem of data detection without errors becomes important to avoid falsely discerning insecurity levels. This paper examines some signal management methods for error free operation of signals from Doppler motion sensors and camera image data. Amongst them this is considered the technique for encoding data using the Hilbert Transform. This is better for motion sensors with a few extra bits of transmission enabling error free reception at the monitoring computer station. Also considered are wavelet compression and transmission for remote site image data using spread spectrum.**

Keywords: *Home security system; Hilbert Transform for data encoding; Wavelet transform; Spread spectrum.*

1. REMOTE SENSOR SURVEILLANCE-METHODS AND PROBLEMS

Now-a-days, it is common to have a central monitoring system of remote surveillance doing the job for many a home or industry floor. The monitoring system itself has to be securely located in a place difficult to identify. It deals with the problem of receiving inputs from a multitude of signal sensors, from all its customers, periodically. It has to process them despite any transmission system noise and identify events of insecurity as well as send suitable commands to take proper action [1]. In this, there are two categories of security levels: i) totally un-manned and uninhabited; ii) partially manned by sentinel.

The methodology and implementation of security monitoring systems are varied and wide. On the one hand, there are various methods of sensing the internal environment to be monitored employing whatever sensors that would fit the environment best. On the other, there are different approaches to prevent intrusion, by simulating the presence of human existence in an unmanned area. When many sensors and cameras are used for many rooms and areas, the quantity of signals become large; their continuous monitoring over extended periods of time render the data manipulation large and extensive. Among these are the several CCTV, metal detectors, burglar alarms, fire alarms, access control and so on [2].

While new techniques in sensors have brought forth more and more components for the security environment, there have also been incidents of pilferage, interference and mal-operation in many sites leading to failure of the entire monitoring system. Attacks against security have been very many. Therefore, in this write – up, we first cite the sensor techniques and then concentrate on how the sensor information is getting immense when a single monitoring station caters to many sites which have entrusted their security problems to it. With such a continuous flow of information to the processing station, data can be interrupted by hostile elements and it can gain errors. Erroneous information in security monitoring would invariably provide false alarms. So much so, it becomes necessary to confirm the validity of such communicated data while also being able to correct errors. In this context, methods conventionally available in data communication for error detection are not sufficient in this respect because none of these could provide the surety of correctness of data, though they could correct a very small percentage or errors which might be present. Particularly important are the motion sensor data with regard to error detection.

2. TYPES OF SECURITY SENSORS

Security makes use of sensors extensively. Sensors must be reliable and should not provide false alarms and must operate with low power and also from battery in order to provide security signaling even when power is purposely shut down in the area being protected. Vendors [3] supply products for real estate developers and hotel operators a cost effective and reliable way to provide their customers with total control over home or hotel rooms.

To provide security in unattended situations, simulation methods are common. The monitoring systems simulate the presence of inmates for external observers. This is by switching on and off, the several internal lights, issuing noise akin to speech and turning on the television set at intervals, particularly during day time. This means a pre-planned appliance switching control from an external monitoring station in order, to simulate the presence of inmates in an actually vacant home or commercial site. Signal commands sent for activating these simulation systems need to be protected because if they are watched by an intruder, it makes it easy for him to intrude such a site.

Sites hired for security management by the central monitoring station may need activation and de-activation as and when needed. Current technology provides for security systems activated with one or more buttons or with a voice command key pad with select regions of security or levels. The activation can also be made from external through telephony, but this has to be handled with suitable password protection and encryption embedded in it.

There are a variety of signal sensors available in this area. The most common are the switches in the several movement paths, tampering signal detection components for lockable items, door signals, lighting signals. There are motion sensors based on the Infra-red, microwave and ultrasound reflectance principle which are capable of sensing human and only human movement. The three states of any sensor signal would be:

i) inactive
ii) active, indicating a presence of an event (closure, opening) and
iii) failed sensor.

Here again, the characteristics of sensors - the resolution, linearity, sensitivity, response time, offset, range, hysterisis, long term stability, their temporal behavior etc., need reliable proper signal hardware concepts.

For movement of humans in unexpected sites and areas, usually reflectance sensors or infra-red thermal detecting sensors are used.

The path-way switches, the door hinge switches are digital; the motion Doppler signals are analog and need processing further; the mixed signals arise from reflectance sensors with varying threshold depending on ambient lighting conditions. Most of these are available with wireless communication in built. Security monitoring based on audible signals detected through remote microphony [4] is also cited with classification techniques based algorithms for detecting non-speech and speech audio.

There are several issues relating to sensors that need to be addressed today. Sensors should be precise and should be communicative by themselves such as the radio remote transmitting door lock sensor [5]. There are also systems like the Sony's Wireless iPod Whole-House S-AIR Audio System [6]. In this paper, with respect to motion sensors, a new encoding scheme is indicated that enables data to be transmitted with error correction embedded in the method.

With Doppler signal data from sensors, the movement profile of the source, which, in our case, happens to be the possible intrusion in the monitored site, should be identified with high level of probability. The ability of the radar to reject faster than walking speed targets can also be controlled by time constants in the circuit. A typical value of radial velocity is 16 mm/s at the MID frequency of 10.687GHz and this determines the range [7].

3. SIGNAL HANDLING AT MONITORING SITE

Sensor signal processing follows sensor signal conditioning. The question as to which signal sensors would do all detection and processing at site and which ones at the remote monitoring station have to be decided first. For most sensors, it involves advanced signal processing algorithms that go far beyond sensor signal conditioning. Examples are linearization, adaptive filtering, correlation, signal transforms, signal compression, and pattern recognition. While some of these can be implemented easily with analog circuits, digital signal processing does the rest. Signals digitized can be stored for long time for comparative inferences. It is possible to send signals all collectively as and when they are sampled and communicate them remotely to the monitoring site via a communication channel. Or else, the processing can be done by local hardware and only the final components of signal complexes be transmitted to the monitoring site.

Complex systems of security levels are required in the several vulnerable locations. Once a preprocessed signal is available, it is not necessary to perform all further processing in the site itself. Doppler signal processing from at least two different motion sensors would need not merely be a signal indicating a motion that is causing a suspicion, but also the time course of the signal indicating its positive aspect of suspicion.

Image signals are also now becoming more and more important. These are usually transmitted with encryption and compression. They need to reduce the data size and increase the throughput.

4. INFORMATION FUSION FROM MULTI-SITE SENSOR COMPLEX

In areas related to industrial and military applications, there are such systems of information processing, such as the (VMMD) Vehicular Multi-Sensor Mine Detection [8]. Land mine detection sensor development and systems integration are evaluating new technologies, incrementally improving existing technologies, increasing the probability of detection, reducing the false alarm rate, and planning out usable deployment scenarios.

With all these sensors and data, one would think that a body of data would exist from which effective real time algorithms for information fusion or "detection" could be developed. The application of Dempster-Shafer evidential theory for implementing a home security system using sensor data fusion has also been considered [9].

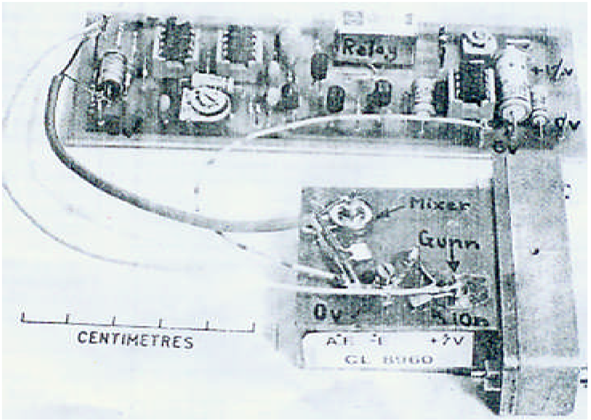


Figure 1 A early bird MID homebrew MID device using Gunn Diode and Mullard CL8960 mixer [5] .

There should be a multiplexed signal processing software, which would handle the several communicated signals from time to time from the customer sites.

Thus, 300 signal bits would need to be processed at least once a few seconds, continuously and with around 100 customers per one network that monitors its area, there would be a signal processing of 30000 bits in unit time slot.

The mathematical processing of the signals would vary from simple to mathematically time intensive approaches, such as complex Fourier transform, for the Doppler signals from motion sensors.

Then, the simulated testing of the sites are also to be included as part of the sentinel system. In this, the site would be simulated with events from a manned or automatic program of closure or opening of proximity switches, movement simulators with lights or infra red beaming lights with motion and so on. The programs for such periodical testing would be part of the maintenance routine which would be run as often as needed or requested by the customer.

So much so, a monitoring station hired to monitor surveillance activity in about 50 sites would be dealing with more than 500 signals over the whole day in sequence; this would require sufficient processing power and software management.

Consider a system with around fifty digital logic outputs from switches and proximity sensors; around a dozen motion sensor input signals processed or given in raw format with just a digitization on an 8 or 10 bit ADC. Combine this with the signals which are created to simulate human presence by the outputs to the lights, the noise speakers, the switching on and off the television or other similar simulated activity, including answering a telephone call from recorded information, all amounting to around 40 approximately.

The above will need a signal vector, which could be having a dimension that would be $50 + 12\times 2 \times 10 + 40 = 330$. The sampling time could be anywhere between hours to a few seconds, depending upon the security system's time resolution. (The Doppler signals are two in number, which are analog but converted into 10 bits digital by the ADC and hence we have $12\times 2 \times 10$). This is just from a single site being monitored for security. There could be many such homes and security monitored sites contracted by the monitoring central agency, which will be therefore continuously receiving information from the several sites all through the 24 hours.

The job management and control strategy as well as emergency functions of the central monitoring system would be to analyze the above data, assuming that there is no loss or corruption in the system used for transmission or any man-made interference in the system of communication. There could also be a protocol communicated via any of the wireless or similar communication systems, including the use of the cellular phone [10].Error free transmission should be the aim, since errors in data, if not identified, will lead to false alarms and disturbances to customers.

5. ENCODING OF SIGNSALS FROM MOTION SENSOR

Motion sensors are a very important of the totality of sensors at sites.

The sensors of the motion detectors quantifies motion that can be either integrated with or connected to other devices that alert the presence of a moving object within the field of view. In Simpler systems, the motion sensors used to be connected to a burglar alarm to alert the security station after it detects motion or the signal used to trigger a red light camera. Amongst these sensors, (PIR) Passive infrared sensors [11] look for body heat. In Some specific cases the signals are usually detected either by all of these and in addition, ultrasound or infra red sensors.

The ultrasonic active sensors send out pulses and measure the reflection off a moving object. The Doppler effect principle is used for ultrasound signal transducers to detect and determine the moving intrusion and hence it is more reliable. To avoid false alarms due to pets, the dual-technology motion detectors are used [12]. In these *PIR/*Microwave combination detectors Pet-Immune functions that allow the sensor to ignore pets that weigh up to 40 pounds or 80 pounds. The IR sensor detects movement by thermal noise from the source which varies much more than threshold with an intrusion in the region of the sensor.

This motion sensor detects IR energy changes in multiple areas and then couples them for extremely accurate detections. Quad-zone Logic provides multi-segmented detection zones over the detection area. It is designed so that a human-size target will normally fill four to eight zones [11], and this will cause an alarm to be generated. Any smaller temperature change (i.e., small to medium-size pets, rodents or moving curtains) only activates one or two zones at the same time, creating a much weaker detection signal.

Using a noise reduction current, the 40 × 40 PIR/Microwave/Pet Immunity motion detector provides high-reliability performance against outside noise, such as electromagnetic interference and especially noise from fluorescent lights, thus solving a problem common to microwave motion sensors. Its Anti-Crosstalk System prevents interference from other microwaves if you have more than one detector in the area**.**

In Motion artifacts, it is known that the signal at an instantaneous time is likely to be corrupted by noise bursts, such as, for instance, a lightning flash (that could also be notoriously simulated by the intruder!). Dual sensors and cumulative signal processing would enable the detection of such artifacts but at double the cost. If the motion is detected without ambiguity, then the remote monitor will call for operator intervention who would then switch on to direct video observation of the site.

With ultrasound motion sensors, we get two signals in quadrature. By taking a combination of these with the Hilbert Transform, we can detect the motion direction. Let $e_{r,}$ e_q be the real and quadrature signals. Taking the Hilbert Transform of the second, we get

$$H\{e_q\} = e_q'$$

Adding and subtracting to e_r, we get the forward and reverse direction time signals.

Therefore, we can transmit the signal e_r and $e_{q'}$ as well as the Hilbert transform of e_r and e_q to the remote monitor for evaluating the movement and assessing direction and further inferences. In this process, we can also detect errors in the signal transmission, as we shall see in the next paragraph, which is an added advantage of this technique.

But transmission of the signal through a medium, say, by some form of radio telephony, is beset with errors of PCM data en-route. Then it means much more for these motion sensors than other direct sensors. A bit error in a stream will intimate a sudden change which will be interpreted as a movement caused by Doppler shift of ultrasound reflected. That is why some form of detecting errors has also to be included.

6. ENCODING OF SIGNALS, NEW METHODS

The Signal *s(t)* that is picked from the sensor is usually digitized and the value of the same is stored in, say, *N* bits, for each sample.

The principle is to take an encoded version of the signal, which uses the signal and its past samples, while transmitting to the distant monitoring site. This encoding is very much like the encoding used for sending GSM signals with convolution encoded data. The system of motion sensor along with this convolution encoding would be built at the site, along with the hardware of the sensor electronics. An example of such encoding by convolving with delayed signal components, would have a block diagram as shown below in a simple scheme with two delays. Actually GSM uses a convolution with a larger constraint length.

The signal is $x_{n,}$ in its n^{th} sample; this is combined with x_{n-2} in order to get y_n

$$y_n = x_n \text{ EOR } x_{n-2} \qquad ...(1)$$

Another bit that is transmitted is $z_{n..}$

$$z_n = x_n \text{ EOR } x_{n-1} \text{ EOR } x_{n-2} \qquad ...(2)$$

At the receiving end the signals are combined by a simple exclusive *OR* and we get the signal $w_{n,}$ which is same as $x_{n-1.}$ Two bits are transmitted for each data bit, see Figure 2. The decoding process at the received monitor would detect and correct errors using Viterbi's algorithm [12]. It is possible to detect errors present and to some extent correct the same.

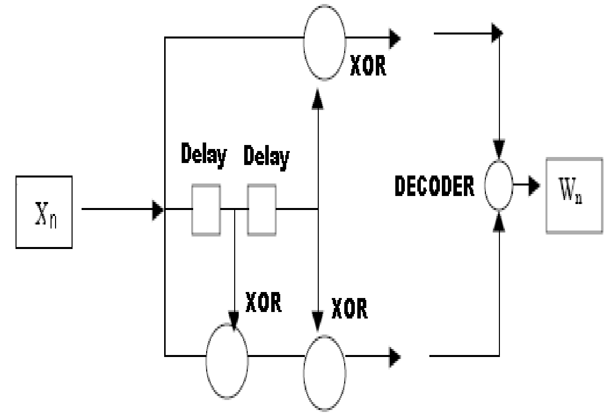


Figure 2 Sensor codes are also convolution encoded and transmitted**.**

The scheme works alright because only detection of errors is more important here rather than its correction.

There is yet another scheme which uses the improved (RS) Reed Solomon coding [13].This is useful for correcting bit errors in data by adding additional bits. The scheme is very complex using Galois field numbers. This has provision only for correcting a small number of bits in a total stream. If n is the data bits, *t* is the number of errors to be corrected, then *n+2t* bits will be total that would be sent by the RS encoder. Usually *t* is very small fraction of the total *n*. For example, the RS (255,235) code has 8% (20/256) of the transmitted message as redundant and it can correct ten errors. This scheme is very popular in general communications. But here, for security applications, this has one drawback. It cannot tell if the data is free from error. Though it can correct *t* errors in a total of *n+2t*, if there were more than *t*, it will give a wrong result. In security data, we can ignore a set of incorrect data rather than having it fully corrected; but we should not be informed with wrong data any time.

In the proposed method, the encoding does not suffer such a limitation. Suppose for the signal *s(t),* after digitization, its Hilbert transform is calculated, this gives the total signal as

$$s' = s_r + js_h \qquad \ldots(3)$$

Here, *s'* denotes the transmitted signal comprising of the real and imaginary parts of the signal. The imaginary part is obtained from the real part through the Hilbert Transform. The same is done using digital data with the formula given in (5). The first term in the above equation is the actual signal and the second is the Hilbert transform.

Therefore, the data bits which are transmitted will be double the actual signal data.

Thus, each bit of s_r is combined with the bit from s_j and a *dibit* is transmitted. This is having the same overhead of 1:2 like the convolutional encoder.

The property of the Hilbert transform is used at the reception data processor. If we take the Hilbert Transform of the latter signal s_j, we get the negative of the real part signal, giving

$$H\{s_j\} = - s_r \qquad \ldots(4)$$

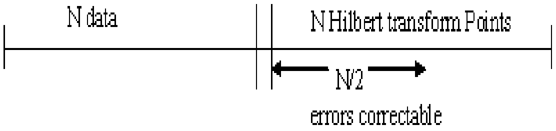


Figure 3a showing the data and parity symbols in the newly developed H.T. coding**.**

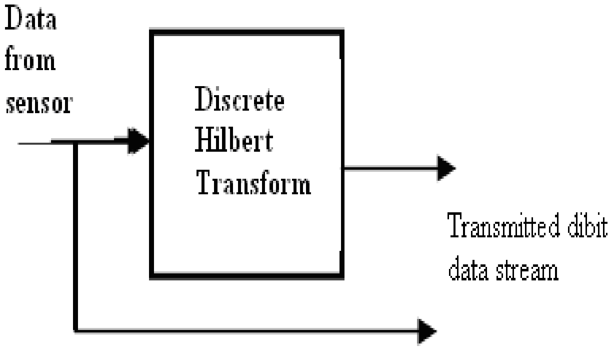


Figure 3b Hilbert transform based data encoding for motion sensors

So, after collecting the data, its Hilbert Transform is also evaluated see Figure 3b.The *N* Hilbert transform values of *N* data are calculable directly as an (FIR) Finite Impulse Response equation and the matrix of calculations for finding the transform can be shown [14] as

$$y'_n = \sum_{j=1}^{N} C_{j+n-1} y_n \qquad \ldots(5)$$

The *y* is data (discrete form) and y' the transform, while the *C* coefficients are a cyclic set of values,

$$\text{Also,} \qquad H\{s_r\} = s_j \qquad \ldots(6)$$

Taking for example, 256 samples of data, the same number of data which are obtained by transforming the former, will make for total data of 512 samples, which is treated as a composite data while sending through the channel. In addition, convolutional data can be used for additional basic bit error correction.

Now, channel noise and interference will produce bit errors on the data, even after correction to the extent possible. Let us illustrate a sample data stream which is just a simple continuous analog signal. The same is digitized and after combining with the Hilbert transformed bits so that the total data stream has two bits for each actual sensor data bit. Suppose, there are random locations of bit errors which cause the value to differ in the received signal at certain sample locations. The signal and its transform are shown in Figure 5 (a-e).

Thus, a check is made on the above data bits by performing digital Hilbert transform, the data part as well as both the transform parts. From this, we calculate the syndrome by the following equations.

$$[a] \rightarrow a_e$$

The data a, gets corrupted as wrong data set a_e.

$$[b] \rightarrow b$$

The other parity part b is the Hilbert Transform of a, which also gets corrupted. Denote H as the Hilbert transform operator,

$$\text{Evaluate } [s] = H\{H(b)\} + [a] - Mean[a]$$

This $[s]$ is the syndrome of the errors received in $[a_e]$.

This is plotted in Figure 4 for this sample data set.

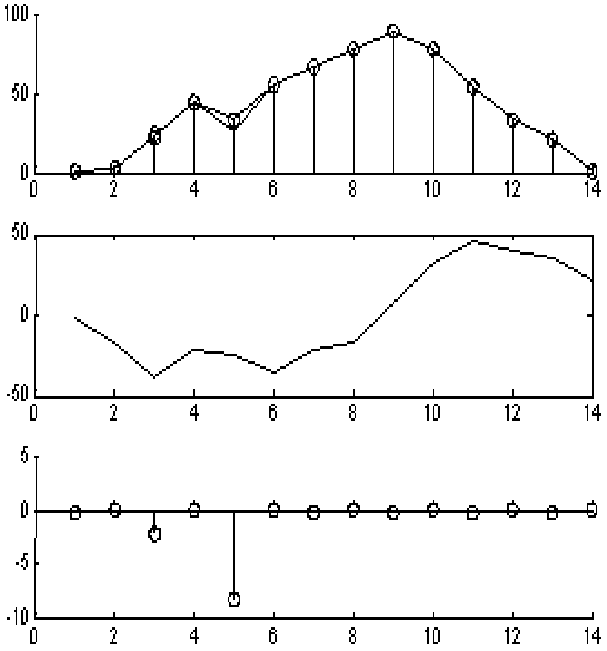


Figure 4 a) Data set plotted, showing error point at (5).
Figure 4 b) Showing Hilbert Transform of data as received.
Figure 4 c) The syndrome shows exactly where errors have occurred.

Thus, it is inferred whether the data is free from corruption in the transmission channel or not. If the equation is non-zero at certain time slots, these time slots could have erroneous data bits. Since some of the data slots are known to be erroneous, ignoring these data bits, the rest of the data is examined for a change from a previous data set for detection of real motion effect. The sensor electronics itself could be using an embedded controller, in which case, the output of the sensor could transmit a combined data set as per equation (3).

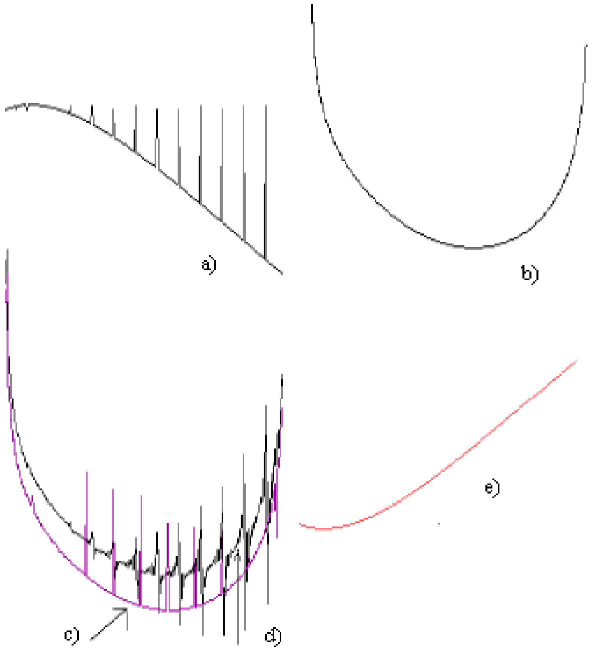


Figure 5 Showing methods of error detection and correction in the Hilbert Transform coding Scheme.
a). Sample segment of signal with bit errors.
b). Hilbert Transform of original signal, as transmitted.
c). Hilbert Transform of received signal part.
d). Hilbert Transform component of the received signal.
e). Transform of corrected Hilbert transform, the negative of signal itself.

Thus, by a suitable encoding technique, it contributes to the overall system reliability.

7. INTRUDER INTERVENTION – IMAGE DATA

Information is often transmitted with a view that security allows a transmitter to send a message to a receiver without the message being detected by an intruder or be disturbed from reception by jamming by a purposely introduced noise signal. Today, any of the mobile communication receivers can be inactivated by properly jamming the select region with high power jamming techniques [15]. A portable cell phone jammer featured by universal and handheld design, could block worldwide cell phone networks within 0.5-10meters, including

GSM900MHz, GSM1800MHz, GSM850MHz/CDMA800 Hz and also 3G networks (UMTS/W-CDMA).

We need definitely techniques to combat jamming in these security maintenance systems, because it is easy for an intruder to jam the signals passed from the monitoring station before he intrudes into the area. It would take time for the monitoring station to understand that jamming has been introduced in a particular home or location, by which time the culprit would have escaped after gaining his objective.

There is no better technique than spread spectrum communication which is proof against jamming. Among the two schemes available, the frequency hopping technique is more useful see Figure 6.

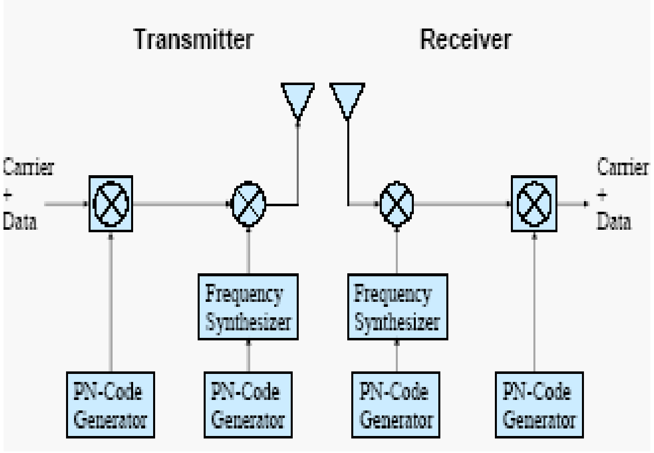


Figure 6 The principle of spread spectrum communication as useful for security monitoring systems.

While the scheme is by itself satisfactory only for en-route jamming, it is not quite satisfactory for jamming from near the site. Further techniques have been investigated by the authors for such a purpose [16].

8 WAVELET TRANSFORMED SPREADSPECTRUM

In Wavelet decomposed spread spectrum, we take the signal *s (t)* and first convert it into wavelets at different frequencies (scales) and time grids as *C (a,b)*. Then, we transmit each of these in different frequencies as a spread spectrum signal. In frequency hopping spread spectrum technique, the signal is sent at different frequency bands as decided by a random PN sequence.

Whatever are the advantages in the conventional Frequency Hopping technique, they are improved when wavelet decomposed signal packets are spread and transmitted [17].

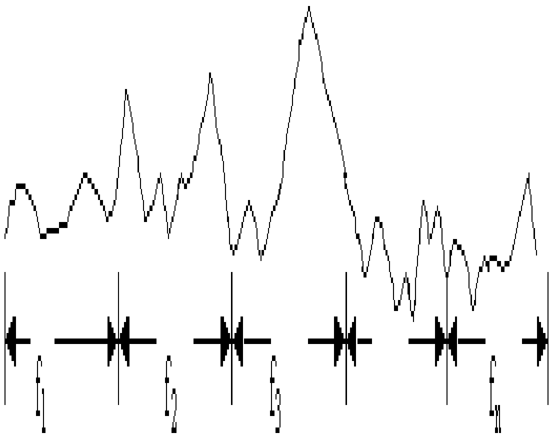


Figure 7 Spread spectrum uses hopping frequencies.

The spread spectrum signal is available at different frequency bands at all times (Figure 7).

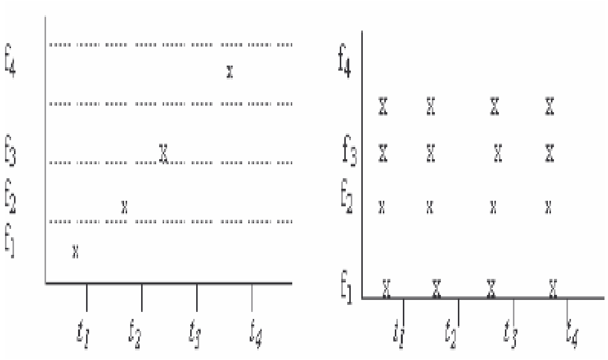


Figure 8a Figure 8b

Figure 8a shows the presence of signals at different frequencies at different at time slots. Figure 8b Wavelet based spreading has signals in all frequency bands at all times.

With the method of sending each wavelet scale in one frequency band, we have the signal available in all the spread frequency bands.

To retrieve the signal back in Figure 8a, we simply take the signal at different time slots, do a filtering with the PN sequence and integrate over a bit time to get the bit value.

If the signal gets jammed in one or more frequency bands, as in Figure 8c (by? code), the probability of error is high at time slots t_2, t_3.

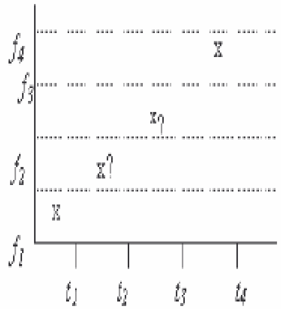


Figure 8c

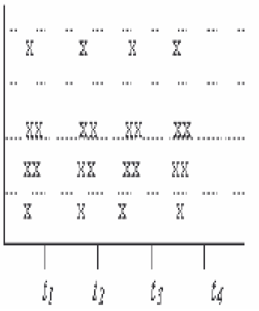


Figure 8d

In Figure 8c The time slots t_2, t_3 are noisy signals caused by jamming. In Figure 8d here, with wavelets at each band, all time slots are noisy in f_2, f_3.

But, in *(d)*, when two bands *(f_1, f_2)* are noisy and jammed, then at all times we get only noisy signals at the two bands f_1 and f_2.

However, the signal is only partly contained in these bands. When we do the wavelet reconstruction of the signal at the t_1 time slot, we get the erroneous output as

$$s(t)_{t_1} = \sum_{a,b} \{\mathbf{dwt}(a,b) + N(b)\}\psi(a,b) \qquad \dots (8)$$

where the *dwt* (a,b) corresponds to the (discrete wavelet decomposed) signal at the band *a* at all time slots (*b*) and N(b) denotes the noise time signal at the time slot *b*, while the function ψ(a,b) is the wavelet function chosen.

8.1 IMAGE DATA FROM LOCATION

The data of an image is coded and sent usually. There was a jamming as shown in Figure 10b, while Figure 10a shows the actual image.

In the method of wavelet based spread spectrum, the image is converted into 2-D wavelet data. The wavelet coefficients for the first level are denoted as:

Ca1, Ch1, Cv1 and Cd1.

These are the wavelet coefficients known as approximate coefficients and detailed coefficients. *Ca1* is the first level approximate coefficient.*Ch1* is the horizontal detailed coefficient; *Cv1* the vertical detailed and *CdI* the diagonal detailed coefficient.

If the image is *256× 256*, the first level coefficients above will each be of size *128 × 128.*

Then, a second level wavelet 2-D *DWT* decomposes the same into a set of further four coefficients

Ca2, Ch2, Cv2, Cd2.

Note that the Ca2 is the second level approximate Coefficient; the others are second level detailed coefficients. The Ca1 is not necessary for reconstruction, if the above four are given.

Thus, the picture or image of 2D DWT coefficients occupy a matrix of size 256 × 256 by stacking the elements of the above *7* coefficient matrices.

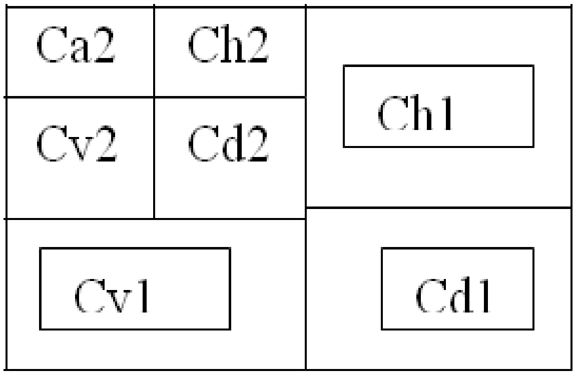


Figure 9 Wavelet coefficients for site image data shown in Segments.

The above matrix is also the same size as the original image for some of the wavelet transforms such as db1 or Haar, but its size will be slightly larger for other wavelets.

The data from the above matrix is got conveniently by tacking the row data and column data into a linear vector. This vector is *65536× 1*. This vector is what is fed to the communication receiver.

Suppose the jamming occurs in this data stream, then what would the above matrix look like (?). It will be jammed at the exact locations as shown in Figure 10b. Thus Figure 10*c* shows this.

Then, this data is used by the receiving communication system. It converts the stream of data thus jammed. Then, the image is got by an inverse 2-D DWT process by equation (8). This gives the matrix image of reconstructed data, though jammed en route.

This image is decoded and shown in Figure 9d There is evidently loss of detail due to smear visible at a few locations in this, but not so many as in the image jammed by the direct jamming as in Figure 10b.

The information as to how much improvement in the process of reconstruction against jamming is given by summing up the pixel errors for the entire image. Thus, if the image original and the image got after jamming shown in Figure 10b is considered, we get an error. But if we do the same with the Figure 10d image, we get error less than the first one.

The ratio between the two errors is varying between 7 to 14 for several images and several kinds of patterns of jamming.

This method, therefore, combines data encryption (because wavelet information is not visible as any meaningful image) as well as error detection. If an image is in error, it might indicate an intruder for instance. Erroneous image received might be subject to false alarming. Here, we can identify errors in data clearly, while maintaining image secrecy [18].

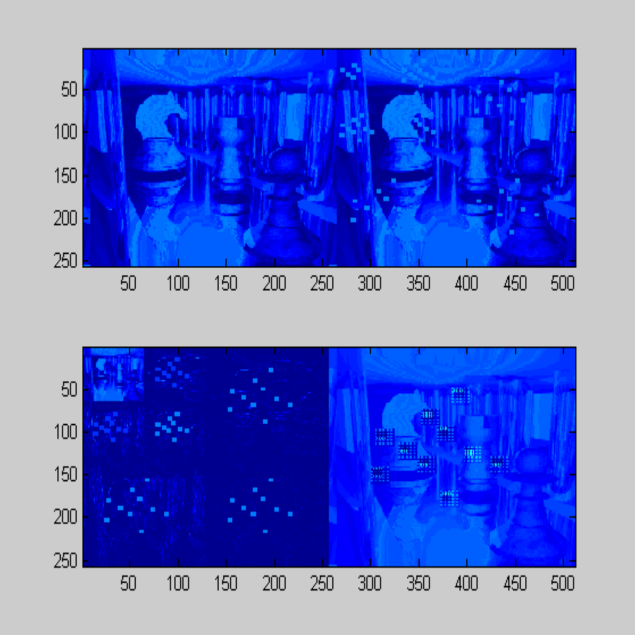


Figure 10a *(Top* left) Shows original image.
Figure 10b (Top right) Shows erroneously received image
Figure 10c (Bottom left) Wavelets
Figure 10d (Bottom right) reconstructed image, showing that data had met with errors.

9. CONCLUSION AND SUGGESTIONS

Data collection from site [19] and transmission for remote monitoring were discussed. Among the several sensors, a method for dealing with motion Doppler sensors was described.

Also, data communication with encoding using the Hilbert transform was considered for detection of errors in data. Clear detection of a motion event due to any intrusion is one of the major criteria in intruder detection. The method based on the Hilbert transform encoding was shown to be better in several respects, compared to the Reed Solomon type of encoding.

Data of images are better handled by wavelets. Spread spectrum communication which is the choice for remote security management can benefit from sending images in wavelet coefficient encoding. This not only does some compression but also provides secure image decoding and error detection.

Security system remote monitoring has been discussed in the context of protecting a living environment so far. Other environments include plant and process units in factories.

In process plant units, data management and control schemes are involved for operating the several plant items like motors, compressors, boiler, reaction vessels and so on. In mechanical plant, there would be robotic assembly lines handling jigs and fixtures. In these situations, security aspects are concerned with data mismanagement by intruders.

With computer based controls all over, the computer data are fed to process units through PLC modules to fix the set values of operating levels of each equipment. It is possible to add units for remote transmission of process variables from every such unit, such as the temperature, pressure, concentration, flow rate etc to the remote station with the plant Identifier code juxtaposed at periodic intervals.

Intrusion herein can be done with undesirable intentions to mal-operate the plant by altering the set values or process variable data. Since most plants of these kinds work under a bus scheme like the Profibus^R, security aspects should be incorporated in these schemes of plant control, with remote monitoring of the data. Schemes for key encrypted handling in such fieldbus control are discussed in [20]. The method uses one of the bits in the protocol to provide for such encryption.

As to the pros and cons of totally local management of security problems versus totally remote monitoring, there are several issues to be thought of. Wherever a security site is fully managed by local system, the problems of communication and jamming by intrusion is avoided. Remote monitoring helps in assessing the integrity of the security system by periodic checks on its inside hardware, which is not feasible with all local monitoring. Remote monitoring is ideal for multiple sites as the cost is distributed among them. Security of several plants located at different sites and handled by a single remotely monitored system is helpful for interactive management among the plants.

There is good scope for development of sensors with built in remote transmission through some form of RF communication with encryption. There is likewise a need for separately developing communication protocols similar to the ones employed for cellular telephony particularly for process plant management. Emphasis can be made on error free data communication even with greater redundancy than used in general communication schemes for absolute error free data collection, which is absolutely important in any security system management.

Future systems should examine and standardize sensors free from artifacts and develop methods for communication. Emphasis must be on detection of errors and possibly correcting them to the extent possible by all known methods of redundant data encoding.

References

[1]. R.Hariprakash, G.Venkat Ramu, T. Rajakumari, S. Ananthi, and K. Padmanabhan: “Some Problems & Methods for Remotely Controllable Automatic Home Security Systems” presented at ICSNC.2008.41, pp. 400-403 , 978-0-7695-3371-1/08©2008 IEEE.

[2]. Security systems and Devices “, controlelectronic.com

[3]. *x*10.com/security Systems

[4]. Abu-El-Quran, A. R., and Goubran R. A: “Security Monitoring Using Microphone Arrays and Audio Classification”, Proc. IEEE Conf., IMTC 2005 pp. 1144-1148, May 2005.

[5]. www.pulseliving.com- Advanced KNX/EIB and Z-Wave automation for homes and hotels.

[6]. Sony AIR-SA20PK Wireless iPod Whole-House S-AIR Audio System smarthome.com

[7]. K. Holford, “Microwave intruder detection”, Wireless World, Vol.86, pp. 34-40.

[8]. Rob Siegel, “Land mine Detection”, IEEE Instrumentation and Measurement Magazine, Vol. 5(4), pp. 22-27, 2002.

[9]. B. Moshiri, A. M. Khalkhali, and H. R Momeni,: **“**Designing a home security system using sensor data fusion with DST and DSmT methods”, Proc. of the 10th IEEE Conf. on Information Fusion, July 2007, pp. 1-6, ISBN: 978-0-662-45804-3**.**

[10]. “GSM phone for remote control”, Elektor India, March 2002.

[11]. “Pet-Immune IR/Microwave Motion Detector Eliminates False Alarms” Optex Product No.: MX-40PI, UPC: 788924082486.

[12]. R.J.Mc. Eliece, R.B Ash and C.Ash, “Convolution encoding and Viterbi’s algorithm”, Chapter-6, Introduction to .Discrete Mathematic, McGraw-Hill Intl., 1989.

[13]. M. Sudan, “Decoding of Reed Solomon codes beyond the error correction bound,” Jour. of Complexity, vol. 13, pp. 180–193, 1997.

[14]. Cizek.V:“Discrete Hilbert Transform”, I.E.E.E. Trans. on audio and Electroacoustics”, Vol.18(4),pp. 340-44, 1970.

[15]. blog.tayx.co.uk/2008/03/10/cell-phone-jammer.pp.682

[16]. T. Rajakumari, S. Ananthi and K.Visalakshi, “Use of Hilbert Transform Data encoding for spread spectrum communications”, Presented at the Intl. Conference on Advanced Telecommunications , (AICT2007) Organized by IEEE at Mauritius , May 13-19, 2007

[17]. Sonia Grgic, Mislav Grgic and Branka Z-Cihlar, “Analysis of image compression using Wavelets,” IEEE Trans. Indl. Elec. 48(3), 2001.

[18]. S. Ananthi, R. Hariprakash, V.Vidya Devi and K. Padmanabhan “Spread Spectrum Communication Using Wavelets of Signal for More Security”, Proc. of the Advanced International conf. on Telecommunications, (AICT/ICIW2006), pp. 87,19-25 Feb’ 06 at Guadelope, (0-7695-2522-9/06 © 2006 IEEE).

[19]. Texas Instruments Data manual: “Converters - Analog to Digital (ADC) 12Bit 50KSPS I2C Lo-Power 8Ch MUX Converters”, SBAS181C – Nov’ 01 - Revised Mar 2005.

[20]. P.Swaminathan, K.Padmanabhan, S. Ananthi and Pradeep.R: “The Secure Fieldbus (SecFB) protocol -- Network communication security for secure industrial process control”, Proc. of the TENCON 2006, Hongkong, pp. 1-4, 1-4244-0548-3/06©2006IEEE.

A Component-Based Approach Based on High-Level Petri Nets for Modeling Distributed Control Systems

Aladdin Masri
Department of Computer Engineering
An-Najah National University
Nablus, Palestine
e-mail: masri@najah.edu

Thomas Bourdeaud'hui Armand Toguyeni
LAGIS – CNRS UMR 8146
Ecole Centrale de Lille
59651 Villeneuve d'ASCQ, France
e-mail: thomas.bourdeaud_huy@ec-lille.fr,
armand.toguyeni@ec-lille.fr

***Abstract*—The evaluation of using *distributed systems DS* in place of centralized systems has introduced the distribution of many services and applications over the network. However, this distribution has produced some problems such as the impacts of underlying networking protocols over the distributed applications and the control of the resources. In this paper we are interested particularly in manufacturing systems. Manufacturing systems are a class of distributed discrete event systems. These systems use the local industrial network to control the system. Several approaches are proposed to model such systems. However, most of these approaches are centralized models that do not take into account the underlying network. In this context, we propose the modeling of the distributed services and the underlying protocols with High-Level Petri Nets. Since the model is large, complex and therefore difficult to modify, we propose a component-based modeling approach. This approach allows the reuse of ready-to-use components in new models, which reduces the cost of development. To illustrate our approach and its reuse capabilities, we will implement it to model the link layer protocols of the norms IEEE 802.11b and IEEE 802.3.**

Keywords-communication protocols; distributed systems; modeling; component-based; Petri nets;

I. INTRODUCTION

Distributed systems [1] [2] are increasing with the development of networks. The development of computer networks has enabled the emergence of new applications benefiting from the power and flexibility offered by the distribution of their functions on different computers. We are interested particularly in this work on the networked control of *manufacturing systems*. Manufacturing systems are a class of *discrete event systems* whose elements are interacting together to build products or to perform services. The concept of *flexible manufacturing systems FMS* has been introduced to develop new manufacturing systems able to produce small or average series of products.

Modeling such systems is very important to verify some properties, especially performance issues. In the literature, many models have been proposed to model manufacturing systems [3] [4] [5]. However, the classical modeling paradigm is generally based on a centralized point of view. Indeed, this kind of modeling does not take into account the fact that the system will be distributed when implemented over different machines, sensors, actors, etc. So, the properties obtained at the design stage are not necessarily guaranteed at the implementation stage.

In addition, the proposed models do not take into account the underlying network and protocols in terms of performance and information exchange. The behavior and design of manufacturing systems are affected by the underlying network features: performance, mobility, availability and quality of service characteristics.

A way to overcome such problems is to model these systems in a distributed way. A distributed system-model offers means to describe precisely all interesting forms of unpredictability as they occur. It takes into account each part of the system, available resources, and system changes together with the underlying network. Once this model is made, its implementation is easier since it has the same characteristic as the desired system. Nevertheless, these systems are complex: they show massive distribution, high dynamics, and high heterogeneity. Therefore, it is necessary to model these systems in a way that provides higher degree of confidence and rigorous solutions.

To cope with this challenge, we propose the use of a *component-based methodology* which is consistent with the principle of distributed systems in which elements are reusable and composable units of code. The component-based approach uses generic, hierarchical and modular means to design and analyze systems. It shows that the system model can be assembled from components working together and the designer needs only to identify the good components that offer suitable services with regard to applications requirements. This methodology allows the reusability and genericity of the components which reduces the cost of the systems development.

In this paper, we propose to model these systems with High-Level Petri Nets which is a powerful tool particularly dedicated to concurrent and distributed formalism, allowing to model both protocol and service components. The work presented in this paper is part of a larger approach on the design of distributed systems by the evaluation, in the design phase, of the impact of network protocols on the distribution of the functions of a distributed system on different computers [6] [7] [8].

The paper is organized as follows. Section 2 introduces the networked control systems and the communication architecture. In Section 3, we focus on the Petri nets modeling formalism. Section 4 gives the properties of modeled components: genericity, modularity, reusability and abstraction level (hidden implementation with connection interfaces). Section 5 shows our methodologies to build the basic components: a top-down analysis methodology and a bottom-up construction methodology. This section presents the component-based approach to develop a library of reusable components based on Petri Nets formalism. At the end of this section, we illustrate the reusability of our components by two examples: IEEE 802.11b DCF and Ethernet. In Section 6, we validate our modeled components by means of simulation to test the correctness of our models. Section 7 presents a case study for modeling manufacturing systems. In this section we show the impact of the underlying network protocols on the services offered by the system.

II. Networked Control Systems

Modern advances in hardware technologies and particularly in communication networks have played a big role in the rapid development of communication networks and distributed control systems.

A. *Networked Control Systems Overview*

Manufacturing systems are a class of discrete event systems. A Discrete Event System (DES) [9] [10] is a discrete-state, event-driven, dynamic system. The state evolution of DES depends completely on the occurrence of asynchronous discrete events over time. Fig. 1 shows the state jumps in a DES from one discrete value to another whenever an event occurs during the time. Nearly all the DESs are complex and require a high degree of correctness. Information systems, networking protocols, banking systems, and manufacturing and production systems fall into this classification.

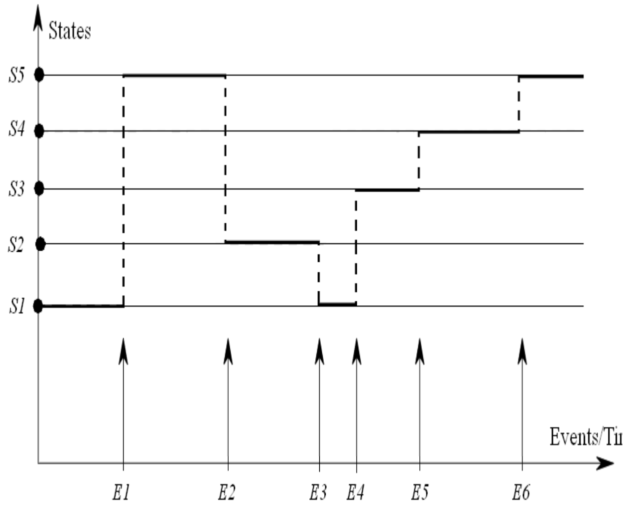


Figure 1. Discrete Event System

In order to improve the adaptability to the market and the quality of manufactured products and to allow their fast evolution, the implementation of *flexible manufacturing cells* is necessary. A flexible manufacturing system (FMS) is a production system that consists of a set of machines connected together via an automatic transportation system. Machines and transportation components such as *robots* are controlled by numerical controllers. In all cases, additional *computers or programmable logical controllers PLC* are used to coordinate the resources of the system.

The cell controllers or computers have a lot of functions and are used to control all the operations of an FMS. The control system manages most of the activities within an FMS like parts transportation, synchronising the connection between machine and transportation system, issuing commands to each machine, etc. *Networking* is extensively applied in industrial applications. The connection of the system elements through a network reduces the system complexity and the resources cost. Moreover, it allows sharing the data efficiently.

Thus, the control of such systems is very important. Nowadays, a controlled system [11] [12] is the combination of sensors, actuators, controllers and other elements distributed around a media of communication, working together according to the user requirements. It is used to manage, command, direct or regulate the behaviour of devices or systems. Combining networks and control systems together facilitates the maintenance of the systems.

The result of this combination is referred to as the networked control system (NCS) [13] [14]. NCS is one of the main focuses in the research and industrial applications. Networked control systems are entirely distributed and networked control system used to provide data transmission between devices and to provide resource sharing and coordination management. These benefits inspired many industrial companies to apply networking technologies to manufacturing systems applications.

B. *Communication Systems Architecture*

Communication systems are designed to send messages or information from a source to one or more destinations. In general, a communication system can be represented by the functional block diagram shown in Fig. 2. The original telecommunication system was developed for voice communications.

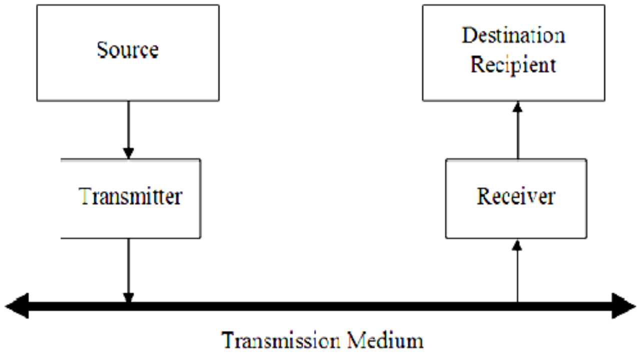


Figure 2. Functional Diagram of Communication System

Today communication networks include all types of voice, video and data communication over copper wire, optical fibers or wireless medium. Networks [15] [16] are organized into a hierarchy of layers where each layer has a well defined function and operates under specific protocols. The number of layers can vary from one network reference model to another but the goal of a layered structure remains common to all models. OSI model [17] is structured in a series of 7 layers, while the TCP/IP model includes only four layers, Fig. 3.

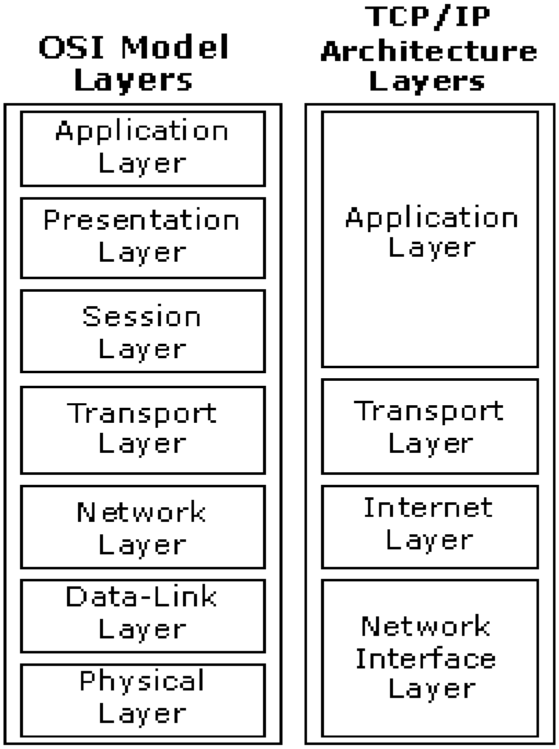


Figure 3. OSI and TCP/IP Reference Models

Each layer consists of hardware or software elements and provides a service to the layer immediately above it. With the advent of Internet, an increasing number of computer networks are now connected.

III. Modeling Formalism

Modeling communication protocols and distributed systems is not new. Several formalisms, simulators and modeling tools exist. However, one of the main challenges in modeling nowadays is establishing a precise relationship within a wide range of available modeling formalisms and comparing their descriptive power and verification capabilities. Thus, we propose to *unify* the modeling of both, protocols and distributed systems, in one formalism. In this way, our work eliminates the need to transform one formalism to another one and so facilitates the modeling process. Since time is an important feature of communication protocols, it is necessary to choose a formalism that allows to properly model the temporal behavior of distributed systems.

A. *Formal Modeling*

The Formal modeling consists of introducing system requirements (cost, security, manufacturing facilities, maintenance, evaluation, reliability and availability) into a small fragment. This introduction of the system requirements must be inside the chosen mathematical framework for the modeling process. The main purpose of a formal modeling is to clarify largely inexplicit information. During the construction of a formal model, many ambiguities must be removed. This consists, in general, of taking a decision. A good model is initially a model that one can understand easily and which can be explained simply. The procedures of verification must be simple and convincing. Several formal tools exist. However, *unified modeling language* UML, timed automata TA and Petri nets are some of the most used modeling formalisms that take into account time factor.

UML [18] is a well-defined, powerful formalism. However, UML lacks one important feature to achieve the desired needs since it has not a formal semantic and hence it is not possible to directly verify *timing requirements* which are *necessary in communication systems*. Timed automata are mainly used to model temporal applications and are not a general purpose formalism.

On the contrary, Petri nets are widely used for modeling concurrent and distributed systems. Many extensions and tools are available, mainly for time, identification of tokens and stochastic issues which are very important issues in communication protocols and services. So, the modeling and integration of all system elements (services and protocols) will be easier (no need to make a transformation). Table 1 shows the most important criteria that we used to choose the modeling formalism.

TABLE I. MODELING FORMALISM

Formalism	**UML**	**TA**	**Petri nets**
Method	Semi-formal	formal	formal
Time modeling	Recently	Yes	Yes
Application	General purposes	Hard timing	General purposes

In computer science Petri Nets are used for modeling a great number of hardware and software systems, and various applications in computer networks. A special advantage of Petri Nets is their graphical notation which reduces the time to learn Petri nets. This formalism has different extensions and tools.

Communication protocols have some specific characteristics and requirements. Thus, the tool selection criteria depends on the following requirements:

- *Time*: Communication protocols are real-time demanding applications. Transmitting and receiving data, accessing the channel, backoff procedure and other needs depend on time. Time Petri nets allow this feature.
- *Headers and Data fields*: Data packets have many fields which may be modeled as tuples. This feature is supported in high-level Petri nets.

- *Probabilistic and Stochastic Properties*: Messages exchanged over the network may be lost or perturbed, bit rate error is a function of noise, etc. The representation of such features can be made by stochastic functions. Stochastic Petri nets are a powerful model to handle such characteristics.
- *Sent and Received Packets*: Messages exchanged over the network require packet identification. Colored Petri nets have been proposed to satisfy this need.

The previous requirements have led us to use the high level Petri nets that combine all the aspects and power of the other extensions in one modeling tool.

B. Modeling FMS with Petri nets

Different modeling formalisms are used to model FMS. Petri nets and Automata theory are of the most used techniques to describe DES. Peterson [19] has introduced Petri nets to model and analyze the manufacturing systems. Other works like [20] have also proposed to model and analyze controllable DES. Later, Lin et al. [21] has introduced a decentralized control model. Zamaï et al. [22] has presented a hierarchical and modular architecture for real-time control and monitoring of FMS. However, newest modeling approaches are diverting towards the distributed modeling for the manufacturing systems [23].

Petri nets have been proposed by C. A. Petri in 1962 in his PhD thesis “Communications with Automata” [24]. Petri nets are a mathematical and graphical tool used for modeling, formal analysis, and design of different systems like computer networks, process control plants, communication protocols, production systems, asynchronous, distributed, parallel, and stochastic systems; mainly discrete event systems.

As a graphical tool, Petri nets provide a powerful communication medium between the user and the designer. Instead of using ambiguous textual description, mathematical notation difficult to understand or complex requirements, Petri nets can be represented graphically. The graphical representation makes also Petri nets intuitively very appealing.

A Petri net graph contains two types of nodes: Places “p” and Transitions “t”. Graphically, places are represented by circles, while transitions are represented by rectangles, Fig. 4. Places and transitions are directly connected by arcs from places to transitions and from transitions to places. A place P0 is considered as an input place of a transition *t* if there is an arc from P0 to *t*. A place P1 is considered as output place of a transition *t* if there is an arc from *t* to P1.

Places can contain tokens represented by dots. These tokens are the marking of places. The initial marking of places is represented in the initial marking vector m0. The graphical presentation of Petri nets shows the static properties of the systems, but they also have dynamic properties resulting from the marking of a Petri net.

As a mathematical tool, a Petri net model can be described by a set of linear algebraic equations, linear matrix algebra, or other mathematical models reflecting the behavior of the system. This allows performing a formal analysis of the model and a formal check of the properties related to the behavior of the system: deadlock, concurrent operations, repetitive activities, etc.

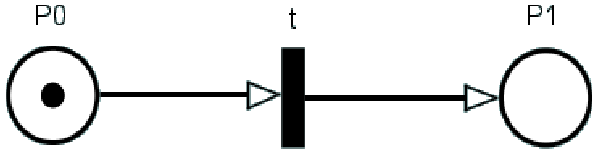


Figure 4. Discrete Event System

C. Proporties of our High-Level Petri Nets

In this subsection we will give a brief definition on the desired high-level Petri nets. This definition is not far from the definition of colored Petri nets [25]. However, we add to this definition a time notation.

Definition: A High-Level Petri Net is a tuple N= (P, T, A, m0, Σ, Λ, G, E, D) where:

- Σ is a finite set of non-empty color sets.
- Λ is a color function, $\Lambda: P \rightarrow \Sigma$
- G is a guard function, G: T → Boolean expression, where:
  $\forall t \in T$: [Type (G(t)) = $B_{expr} \wedge$ Type (Var (G(t))) $\subseteq \Sigma$], where:
  Type is the color type of the guard function,
  B_{expr} is a Boolean function
  Var is the variables of the guard function.
- E is an arc expression function, E: A→E(a), where:
  $\forall a \in A$: [Type(E(a)) = $\Lambda(p(a)) \wedge$ Type (Var (E(a))) $\subseteq \Sigma$], p(a) is the place of arc a.
- D is a delay function, D: E → TS, where TS is a delay associated to the arc inscription with the annotation symbol “@”.

The arc expression function can contain any sign and/or mathematical or logical functions, such as programming language expressions. The delay function can be associated to both output arcs (from places to transitions) and input arcs (from transitions to places).

IV. Componenet-Based Modeling

Component-based engineering [26] has a huge importance for rigorous system design methodologies. It is based on the statement which is common to all engineering disciplines: complex systems can be obtained by assembling components, ideally commercial-off-the-shelf (COTS) [27]. Reusability and extensibility are key factors that contribute to this success and importance. Component-based development aims at decreasing development time and costs by creating applications from reusable, easily connectible and exchangeable building blocks.

In component-based engineering research literature, several approaches [28] [29] have focused on the aspects of the development of components. However, reusing available, ready-to-use components decreases time-to-market for new systems and applications. This may be done by selecting the appropriate components from the available components based on the needs and then assembling them to build a new component system-model.

Different methods of component specification software exist; from the Interface Description Language IDL (Object Management Groups' CORBA, java based components such as JavaBeans and Microsoft's .Net) to formal methods, by design-by-contract methods. Despite their widely difference in the details, they have a common concept: a component is a black box that is accessed through exposed interfaces. In this section we will state more precisely the main features that a component-based method must verify.

A. Genericity

Genericity is used in component-based engineering to reduce time-to-market, and raise productivity and quality in systems development. The term generic component refers to a component that implements a process or part of a process-set and that is adaptable to accommodate different needs. Genericity of a component is based on its independence compared to its type. This is an important concept for high-level methods because it can increase the level of abstraction of these methods.

A generic component can be seen as a parameterable element. Parameters should be specified and a specific version of the component (an instance) will be created and used. Another advantage is that a generic component can be represented as a *generic factory* that will create as many components as necessary for the application. Thus, the main objective of genericity is to integrate the component-based approaches with the technical approaches.

B. Modularity

Modular models are easier to design compared to similar complex models. "*Modularity is having a complex system composed from smaller subsystems that can be managed independently yet function together as a whole*" [30]. The objective of modularity is the ability to identify homogeneous, compatible, and independent entities to satisfy the needs of a system or an application. In many domains, modularity is essential to manage the design and the production of complex technology. Modular design aims at organizing complex systems as a set of components or modules. These components can be developed independently and then joined together.

The decomposition of a system model into smaller modules has the following advantages:

- A modular model can be very near to the real system, since it reflects the hierarchical structure inherent to the system.
- Components which are too complex can lose some of their details and their interactions can be confused. A component can be divided into smaller components until each module is of manageable size.
- It is possible to concentrate on each component as a small problem.
- Modular model allows testing each component separately.
- Implementation changes and corrections on simple components can be done easily.
- Documentation in modular structure becomes also easier.

C. Reusability

The implication of reusability is that the available components must give enough information to ease the assembly of components into a new system [31]. The information must include dependency and configuration information. To take sound decisions about selecting and reusing components, the following information is required:

- Operational specification: the semantic interaction of the component,
- Operation context: where and how the component will be used,
- Non-functional properties: describe the properties such as performance, security and reliability,
- Required interfaces and resources: the functionality and resources needed by the specified component to execute its main functionality.

Since all real systems are made of components, component-based systems are built of multiple components [32] that:

- are ready "off-the-shelf," (generally called "Commercial Off The Shelf"),
- have significant combined functionality and complexity,
- are self-contained and can be executed independently,
- will be used "as is" without modification,
- must be combined with other components to get the desired functionality.

All these benefits and more led us to use the component-based approach to model the distribution of manufacturing systems and the underlying protocols. The reuse of components is very important in the modeling level since most of the system parts and machines are the same. In addition, protocols share many properties. With the reuse of already modeled components, the time and modeling-cost are reduced. As we can see in Fig. 5, models are sharing some properties (the triangle). Once this part is modeled, it can be reused in any model that has a need to such component.

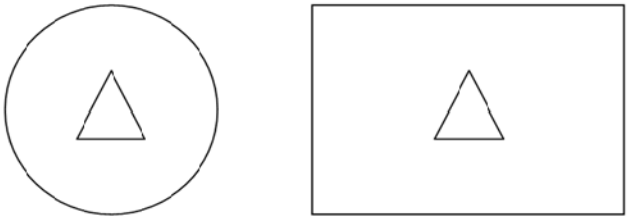


Figure 5. Basic Module Reusability

D. Components Abstraction

The modeled components are seen as black box where the internal functionality is hidden, while the interfaces represent the service that can be offered by this component. Every component or module is characterized by its internal hidden behaviour. Its interfaces are chosen to reveal as little as possible about its inner implementation.

Components abstraction is useful for reducing the design complexity by decomposing a problem into connected components. Abstraction (or specification) describes the functional behavior of the components, i.e. components are considered to be specific to an application. Abstraction focuses on the important characteristics of component upon the designer point of view. This definition supports the abstraction of data, hiding internal function, reusability and self-contained component behaviour descriptions.

Thus, during the design of components we must focus on well-defining the service offered by the component at its interfaces and the parameters that can be adapted to the application requirements, rather than spending the time on describing its internal behaviour. This can be achieved by giving appropriate names to the interfaces and parameters and documenting these interfaces and parameters.

E. Components Interfaces

Components can be built according to the needs of the user and different requirements and points of view. However, these components are characterized by:

- The service they offer: each component has its own functionality and service. The resulting of this service depends on the parameters and value given to the component.
- The hidden implementation: the service and functionality are hidden. However, the designer has the access to the internal code but there is no need to modify the code.
- The interfaces: to access the component service or to connect the components, interfaces are used. Several modes of connection between the different components in the model can be defined.

The component interfaces declare the services that a component offers. They are used as an access point to the component functionality by other components. Since we use Petri nets to model the different component behaviors, we used *places* to be the input interfaces of components and the output interfaces are *transitions*. The input interfaces (places) receive as many tokens as the producer components. The output interfaces (transitions) generate as many tokens as the consuming components, Fig. 6.

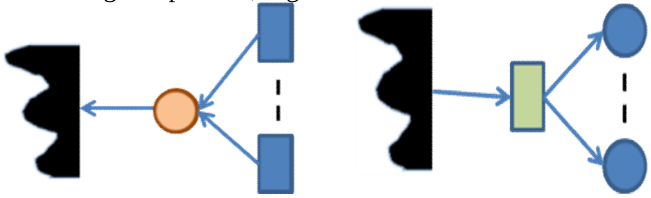


Figure 6. (a) input interfaces (b) output interfaces

This choice is coherent with the traditional way to model asynchronous communication between processes modeled by Petri Nets. Moreover it guarantees the genericity of the components and facilitates the connection between the different components.

The connection between interfaces of two blocks can be 1-to-many, many-to-1 or 1-to-1. As an example, Fig. 7 shows a many-to-1 and a 1-to-many connections. To illustrate the interest of this choice of interfaces, let us consider the modeling of workstations connected to a communications bus. A many-to-1 connection is used to connect workstations output transitions to a medium input place since workstations put their data on the medium only. A 1-to-many connection is used to connect the medium output transitions to workstations input places, since all the workstations can see the signals propagating on the medium.

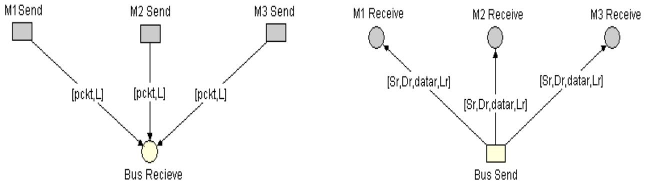


Figure 7. (a) input interfaces (b) output interfaces

This approach is very useful to deal with the complexity due to the size of a system. Indeed, if one has already a model of some workstations connected on a bus and one wants to increase the size of its model, the connection of new workstations can be done easily just by adding an arc between the output transition of the bus model and the input place of the station model. . So this does not require any modification of the bus or the workstation component. Conversely, if the transitions are used as input interfaces and places as output interfaces, the addition of a new workstation would need to add a new token in the output place, and hence modify the internal code, so we loss the genericity.

V. MODELING COMMUNICATION PROTOCOLS

In our approach, we want to model reusable components. In this section, we will build the components that will be used to model the communication protocols. The modeling will be hierarchical since we build first the basic components. Then, with these components, we construct composite-components.

Before starting the construction of modeling components, we will analyze the data link layer protocols that we are interested in this work. These analyses will help to identify the basic common behaviors of the different protocols that lead to definition of the basic components. These basic components are the initial bricks of the library that will serve to model all the complete behavior of the different protocols.

A. A top-down analysis methodology

To build the basic components one must identify these components to be reused in different models. Since we are interested in manufacturing systems, the analyses will be made at the Data Link Layer protocols. The *Data Link Layer DLL* is the second layer in the OSI model. The data link layer is often split in two sub-layers: the *logical link control LLC* and the M*edia Access Control MAC*, Fig. 8.

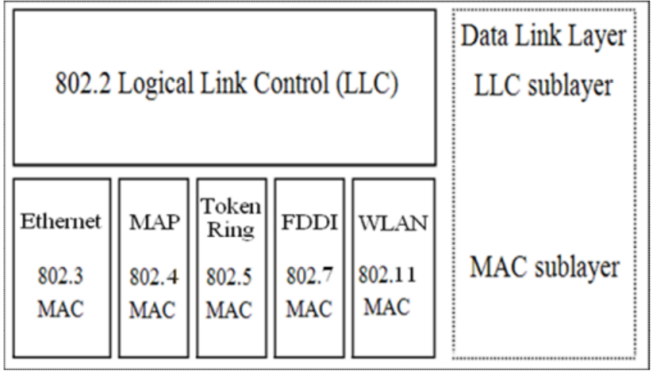


Figure 8. IEEE MAC Sublayer

The MAC sub-layer provides hardware addressing and channel access control mechanisms that enable hosts to communicate. Different MAC protocols are used. The data link layer defines most LAN and wireless LAN technologies: *IEEE 802.2 LLC, IEEE 802.3 Ethernet*, *IEEE 802.5 Token Ring, FDDI and CDDI*, *IEEE 802.11 WLAN.*

The next step is to define the protocols that have the same functionality. Here, one can find two protocols Ethernet IEEE 802.3 [33] and wireless IEEE 802.11 *Distributed Coordination Function DCF* [34] protocols that share the carrier sense multiple access CSMA procedure [35] to send the data over the shared medium. Finally, one must find the common behaviors to associate basic components to it. The result of these analyses is three basic common elements:

1) Channel check:

A workstation attempting to send data must at first check if the channel is free or not. Ethernet uses the CSMA/*CD* Protocol. Here, CD means collision detection. The workstation must check if the channel is free for a period of 9.6µs first, then it starts its transmission.

The IEEE 802.11 DCF uses the CSMA/CA protocol. Here CA means collision avoidance. To use the network, a workstation must before check if the channel is free for more than a period of time called Distributed Inter-Frame Space DIFS, Fig. 9. If so, the workstation starts a random backoff before starting its transmission. If the channel status is changed in both Ethernet and IEEE 802.11 deferring and backoff times, the workstation must restart the process of sensing the channel.

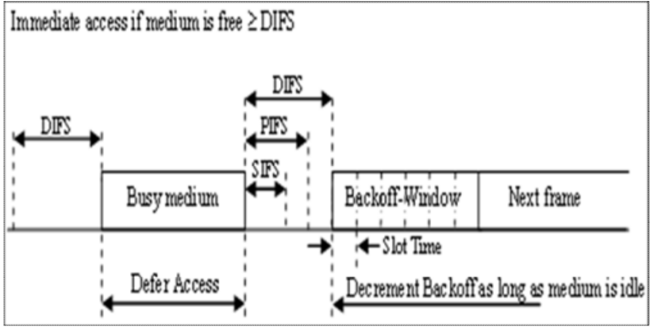

Immediate access if medium is free ≥ DIFS
DIFS
DIFS
PIFS
SIFS
Busy medium
Backoff-Window
Next frame
Slot Time
Defer Access
Decrement Backoff as long as medium is idle

Figure 9. Channel Access in IEEE 802.11 DCF

2) Sending and Receiving: Data, Acknowledgments and JAM:

Workstations send and receive packets. These packets can be data packets, acknowledgment packets or JAM frame (a 32-bit frame, put in place of the correct MAC CRC). In Ethernet networks, workstations receive either a data packet or a JAM after a collision. The destination workstation does not need to send an acknowledgment to the transmitter at the MAC layer. However, in wireless LANs, the destination workstation must send an acknowledgment to the transmitter after a successful reception of a packet, Fig. 10. Otherwise, the transmitter will consider that its packet is lost or a collision has occurred, so it will retransmit this packet causing an extra load on network worthlessly.

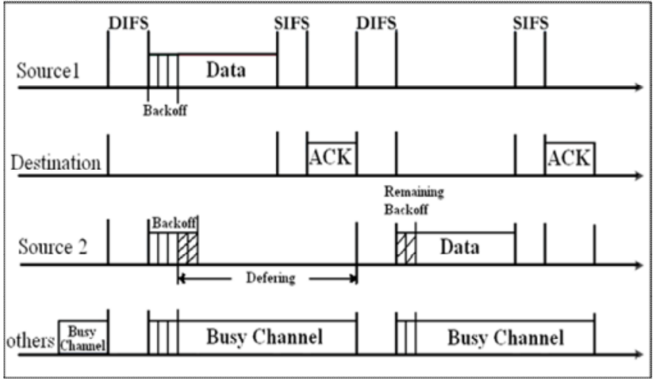


Figure 10. Backoff mechanism in IEEE 802.11 DCF without RTS/CTS

On the other hand, to send data, workstations need only to put the destination address in the packet. Since the medium is shared in most LAN technologies, all the workstations will see the packet. However, only the workstation that has the destination address reads the packet and the others will either forward it, or drop it.

3) Random and Binary Exponential Backoffs

In communication networks errors can occur. This is due to many factors like the surrounding environment, noise and interference, or because of collisions. Ethernet and IEEE 802.11 networks use the channel check and the inter-frame space to decide the medium access. Thus, collisions may occur when more than one workstation transmit on the shared medium at the same time. In Ethernet, the maximum time needed to send the first bit from one end to the other end of a 10BaseT medium is 25.6 µs. During this time, (an)other workstation(s) may attempt to send its data, as that the channel is considered as free.

As a result, a JAM signal is propagated over the shared medium informing the occurrence of a collision. Each workstation concerned by a collision starts a binary expositional backoff procedure, called *BEB*, to decide when it can do a new attempt to access the medium. The BEB algorithm computes randomly a waiting delay that increases with the number of the attempts *Tn* of the workstation.

At the beginning *Tn* equals zero (See Fig. 11). Each time a collision occurs, the workstation increments *Tn* counter until it reaches 15. Before trying to transmit its data again, the workstation starts a BEB by taking a random value between 0 and 2^X and multiplies it by 51.2 µs, where:

$$X = \begin{cases} Tn, & \text{if } 0 < Tn \leq 10 \\ 10, & \text{if } 10 < Tn \leq 15 \end{cases} \qquad (1)$$

This helps in decreasing the possibility for a collision occurrence. In case of no collision, the workstation continues transmitting and when it is done it leaves the channel. However, If *Tn* reaches 15, (the load on the channel is very high), then the workstation aborts its transmission and tries it again later.

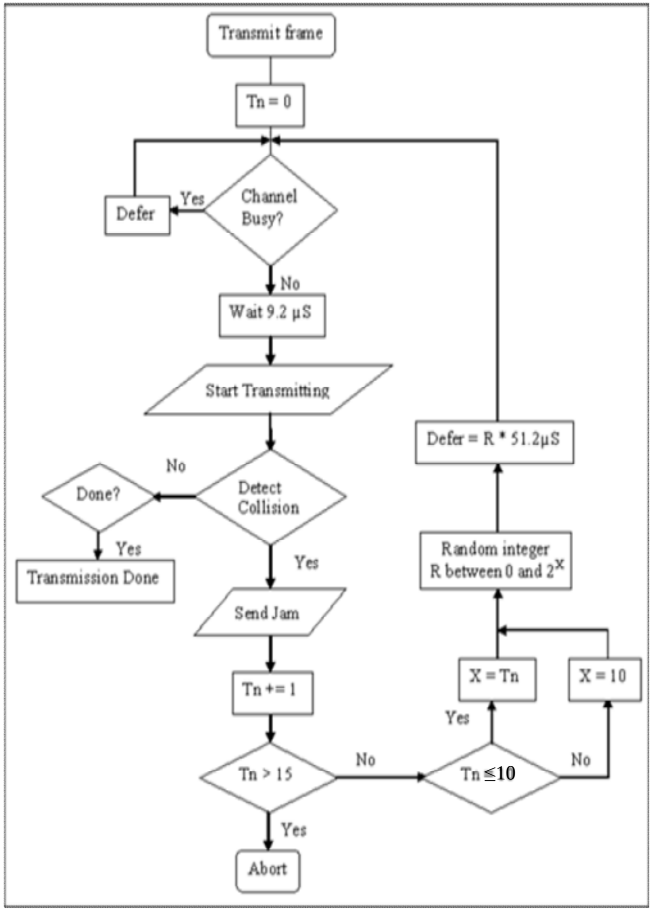


Figure 11. Transmission in Ethernet

In wireless LANs, after a collision, no JAM signal is sent. However, if the workstation does not receive an acknowledgment after a period of time equals to Short IFS *SIFS* (Fig. 9), it considers that a collision has occurred and starts a backoff procedure. For each retransmission attempt, the backoff grows exponentially according to the following equation:

$$ST_{backoff} = R(0,CW) * Slot\text{-}time \qquad (2)$$

Where:

- *ST* is the backoff time.
- *CW* is the *Contention Window.*
- *R* is a random function.

In general, the initial value of CW (CW_{min}) is 16. After each unsuccessful transmission attempt, CW is doubled until a predefined maximum CW_{max} is reached (often 1024).

There are two major differences between Ethernet and IEEE 802.11 backoff processes:

1- The wireless LAN starts a backoff procedure even at the first attempt to send its data (Fig. 10), while Ethernet does not. This is one of the mechanisms used to implement the Collision Avoidance feature of CSMA/CA.

2- Ethernet starts its BEB algorithm after a collision (without conceding the status of the channel) and then restarts checking the channel to send its data. While in IEEE 802.11, the workstation checks first the channel status and then it decrements its backoff by:

$$R = \begin{cases} R - 1, & \text{if the channel is free during 1 time slot} \\ R, & \text{if the channel becoms busy} \end{cases} \qquad (3)$$

The design of CSMA protocol offers fair access in a shared medium. This means that all the workstations have a chance to use the network and workstations cannot capture the channel for ever. The remaining value of R is reused after the channel status becomes free for more than a DIFS period. The workstation starts sending its data when R equals zero, Fig. 12.

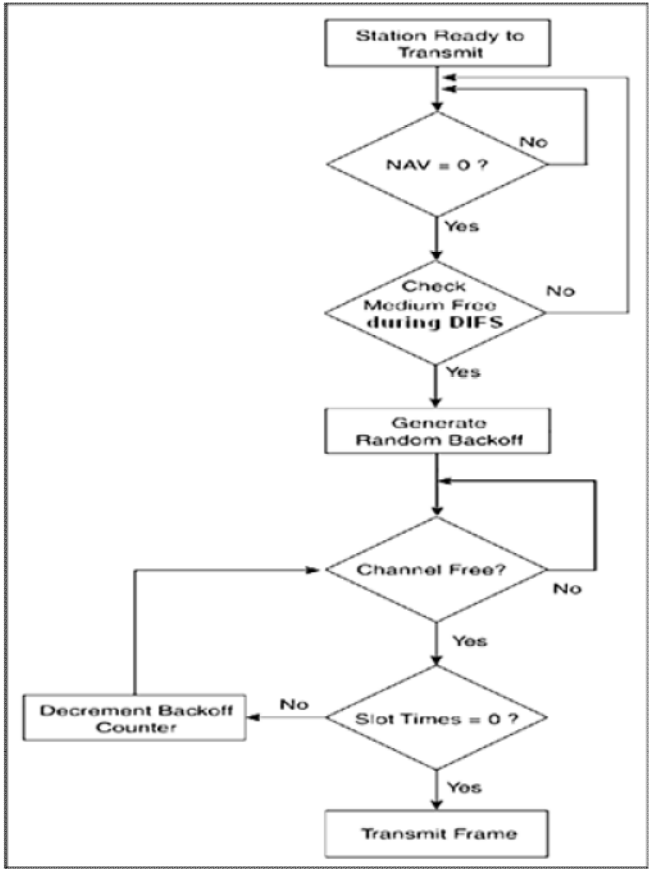


Figure 12. Medium Access Process for 802.11 Protocol

B. A bottom-up construction methodology

As one can see in the last subsection, three elements are in common. These elements can now be used to model the basic components.

1) The channel-check component

Fig. 13 shows a channel check component. Elements in light gray represent the places and transitions used to build the component. Elements in dark gray represent the interfaces of the component. Initially, the channel is idle for all workstations. This is represented by a token in place "Idle". A workstation that wants to send data (a token in place "Data send") must first check the channel.

In wireless LANs, the channel must be free for a period more than DIFS, while in Ethernet, it is 9.6 μs. This is represented by the '@t' at the arc between place "Idle" and transition "TF" (t' equals 9.6 μs in Ethernet and 50 μs in 802.11b). The workstation must wait before it starts transmitting, represented by a token put in place "sdata". In Ethernet the wait "@t" equals to 9.6 μs, while in 802.11 it is equal to random value between CW_{min} and CW_{max} slots time. Place "Backoff/Deferring Time" and transition "FC" is used to decrement the backoff in wireless LAN, while for Ethernet, it can be left as it is in the figure (no dependence to that transition in the model).

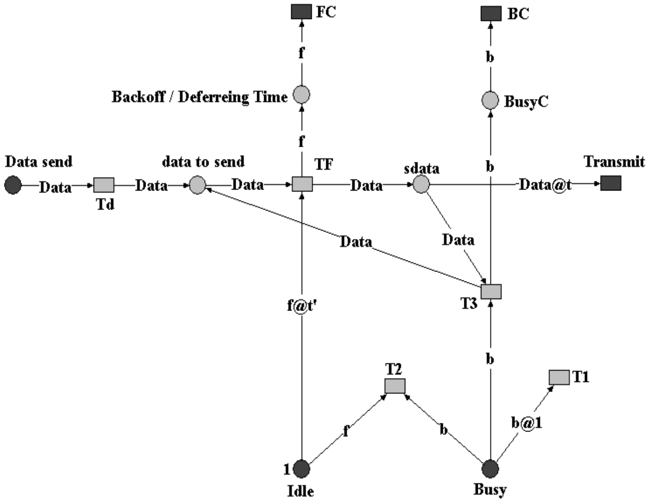


Figure 13. Transmission in Ethernet

Consequently, if the channel status is changed (a token is put in place "Busy"), the workstation can be in one of the following states:

- It is the transmitter (there is no more tokens in place "sdata"), then nothing is changed and the token in place "Busy" is consumed by transition T1;
- It attempt to send or it has no data to send, then T2 is fired;
- It is in the backoff/deferring phase, then T3 is fired (the workstation rechecks the channel again) and a token is put in place "BusyC" to stop decrementing the backoff. Hence, in wireless LAN, the workstation stops decrementing the backoff, but it keeps its remaining value.

In the three cases the channel status is changed from idle to busy.

Initially, this component has one token with value 1 (representing the free channel) in place Idle. The use of this component is possible in any protocol that demands the sensing the channel before transmitting data. It represents also the status of the channel free or idle. Let us notice here that, for genericity, we use two parameters t' and t to define the delay on the arc Idle-FT and arc Backoff/Deferring Time-Transmit.

2) Receiving and sending ACK component

Workstations receive two types of packets: data packet and ACK/JAM frames. In Ethernet network, no acknowledgment is sent after the reception of packet. Therefore, the received packet can be either a data packet or a Jam frame. While in wireless LAN, the received packet is either a data packet or an acknowledgment frame.

Fig. 14 shows the receiving and sending acknowledgment component. One assumes that a token is put in place "Receive". The fields of the token represents: the source address "Sr", the destination address "Dr", the received data "rdara" and the last field represents the lengths of the packet. The workstation checks at first the destination address "Dr" of the packet. The guard condition on transition "Address" checks if the received packet belongs to this workstation, a token is put in place "Data?". Otherwise, the token in place "Receive" is eliminated by transition "Drop". Hence, for simplicity, "Dr==1" is considered as the own address of the workstation, while "Dr==0" is used to represent the multicast or JAM frame reception.

Next, the guard condition of transition "ACK/JAM" is used to check if the received frame is an ACK frame or a JAM frame (for Ethernet only). The "abc" in the guard can be modified according to the needs of the designer and the type of network. However, if the received packet is a data packet, transition "DA" is enabled. This transition is fired after a time equals to the time needed to receive the packet modeled by the "@time(Lr)" at the outgoing arc. This "@time(Lr)" is a function that returns the time corresponding to the length "Lr" of the packet.

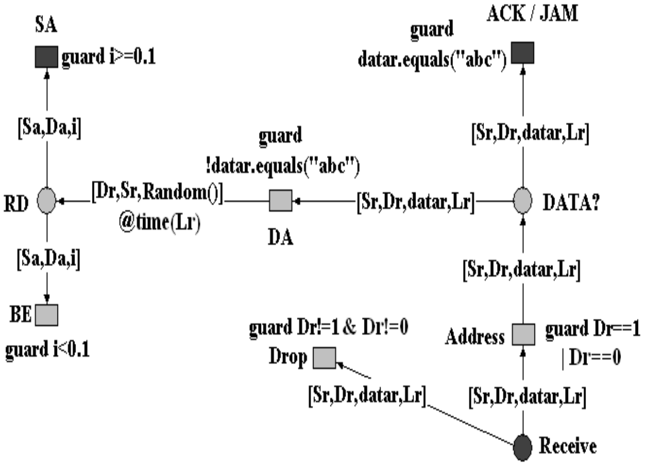


Figure 14. Receiving and Sending ACK Component

Let us notice here, the functions dynamicity can be used to model mobility of a wireless networks nodes. This can be done since the bit rate is a function of the signal strength and that the signal strength is a function of distance. This means if the source knows the location of the destination, then the distance can be computed, and hence the time needed to send a packet is determined.

The last step is to represent the bit rate or receiving errors. The random function *Random()* is used to generate a random variable i. Assuming that the bit rate error is less than or equal to 10% of the transmitted/received packets. So, if the value of i is less than 0.1, then the packet is discarded (the token in place RD is consumed by transition “BE”). Else, the packet is received correctly and then an acknowledgment is sent, by firing transition “SA”. This interface can be left unconnected in Ethernet. As we can see in Fig. 14, the modification of tuples can be done easily, just by modifying the arc inscriptions according to our needs.

As one can see, this component has an important functionality since it is used to identify the received data (own or not), the type of the received data (JAM, ACK, data frame) and the process of sending an acknowledgment after a successful reception. Thus the use of this component is possible for the protocols demanding the identification of data and the send/receive process.

3) *Backoff / BEB component*

The third component is the backoff / BEB component shown in Fig. 15. As we can see in the figure, retransmitting the packet depends on the value of n, (transitions T6 and T7). If the packet is correctly sent/received (a token is put in place “Done”), then n is reset to z (0 for Ethernet and 1 for wireless), for the next attempt to transmit, place N. However, the component inscriptions depend on the type of the network. As an example, Table II shows the differences between Ethernet and IEEE 802.11b networks.

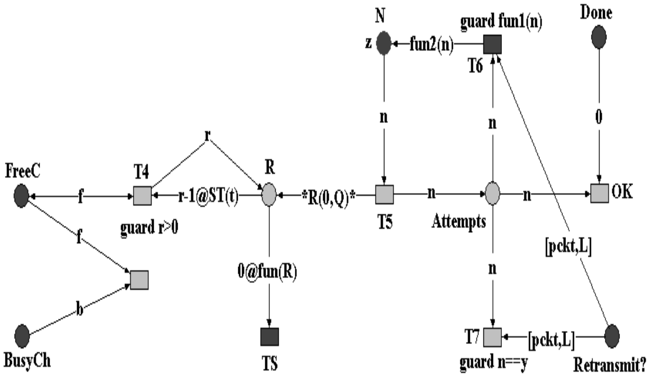


Figure 15. Backoff / BEB Component

In addition to Table II, in Ethernet, places “FreeC” and “BusyCh” are not used (they can be left as it is), since the backoff decrement in Ethernet does not depend on the status of the channel. While in 802.11b, this interface is very important in decrementing the backoff each time the channel is free for a slot time or the backoff is conserved if the channel status is changed to busy.

TABLE II. DIFFERENCES BETWEEN ETHERNET AND IEEE 802.11B NETWORKS

Variable Value	Ethernet	IEEE 802.11b
fun1(n)	n<15	n<33
fun2(n)	n=n+1	n=n*2
y	16	64
z	0	1
R(0, Q)	random(0, 2^X), X depends on n	random(0, CW)
Fun(R)	R*51.2µs	0
ST(t)	0	Time slot (20µs)

The firing of transition TS represents the (re)transmission allowance of a packet (backoff equals to 0). The backoff component is useful for the protocols that may need a specific timing procedure since it can be related to another components (which the case of wireless: by checking channel always) or just for standalone use

C. *Application protocols*

In this subsection, we will illustrate our modeling approach through two examples: IEEE 802.3 Ethernet MAC protocol and IEEE 802.11 MAC protocol because both protocols are based on CSMA. One of the objectives is to illustrate the advantage of having generic components and the hierarchical composition that allows building composite-components.

1) *Modeling an Ethernet workstation*

Ethernet is the most widely used LAN technology in the world. Ethernet was designed at its beginning at the Xerox Palo Alto Research Center PARC, in 1973. The used protocol differs from the classical protocols like token control, where a station cannot send before it receives an authorization signal, the token. With Ethernet, before transmitting, a workstation must check the channel to ensure that there is no communication in progress, which is known as the CSMA/CD Protocol.

Fig. 16 shows the detailed and complete module for the Ethernet workstation. As one can see in the figure, the three components: Backoff component, Channel Check component and Receive/Send component are reused to build the workstation. To complete the model and to bind the used components together, some additional places and transitions (in white) are used to answer the specification of an Ethernet workstation.

In the figure, one can see that five interfaces were not connected:

- The “FreeC” and “BusyCh” interfaces of the Backoff component, and the FC and BC interfaces of the Channel Check component, since Ethernet workstations decrement their backoff without the need to check whether the channel is idle.
- The SA interface of the Receive/Send component, because in this part we do not model the service offered by an Ethernet workstation

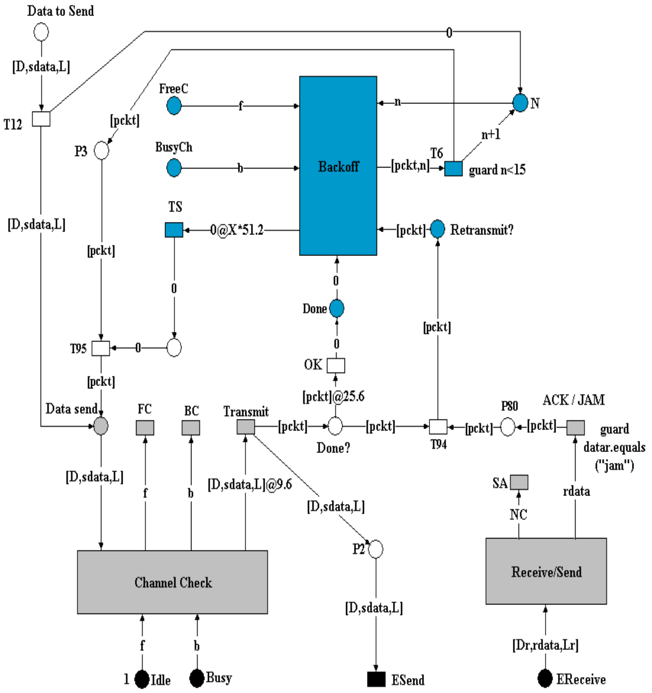


Figure 16. Composite design of an Ethernet Workstation Component based on Generic Basic Components

An important notice is that the whole component can be reused as one component for the Ethernet workstation to build a complete Ethernet network. In other words, this new component is seen as a composite-component with the black places and transitions as the interfaces of this new component.

2) *Modeling a 802.11b DCF workstation*

Wireless technology has become popular to access to internet and communication networks. The IEEE 802.11 offers the possibility to assign part of the radio channel bandwidth to be used in wireless networks. The IEEE 802.11 protocol is a member of the IEEE 802 family, which is a series of specifications for local area network technologies, (Fig. 8). IEEE 802.11 is a wireless MAC protocol for Wireless Local Area Network WLAN, initially presented in 1997. The IEEE 802.11 standard defines Medium Access Protocol and Physical (PHY) frameworks (layer 2 in the OSI model) to provide wireless connectivity in WLAN. This independence between the MAC and PHY has enabled the addition of the higher data rate 802.11b, 802.11a, and 802.11g PHYs. The physical layer for each 802.11 type is the main differentiator between them. However, the MAC layer for each of the 802.11 PHYs is the same.

Many other 802.11 variants have appeared. For instance, in 2004, the 802.11e was an attempt enhancement of the 802.11 MAC to increase the quality of service. The 802.11i and 802.11x were defined to enhance the security and authentication mechanisms of the 802.11 standard. Many other variants exist like 802.11c, 802.11d, 802.11h. The IEEE 802.11 MAC layer defines two coordination functions to access the wireless medium: A distributed coordination function DCF and a centralized coordination function PCF (Point Coordination Function).

Fig. 17 shows the detailed and complete module for the DCF IEEE 802.11b workstation model by the reuse of ready-to-use components designed from the previous sections. The workstation sets the value of N to 1 (place "N"), sense the channel (transition "TF"), sends its data (place and transition "Send") and waits for an acknowledgment (place "Wait"). If no acknowledgment is received during the SIFS period or 10µs, Transition T11 will fire putting a token in place "Retransmit?" to check if the packet can be retransmitted (transition T6) or not (transition T7).

As one can see in this figure, all the components are reused to compose the workstation module. All the interfaces were also used in this module.

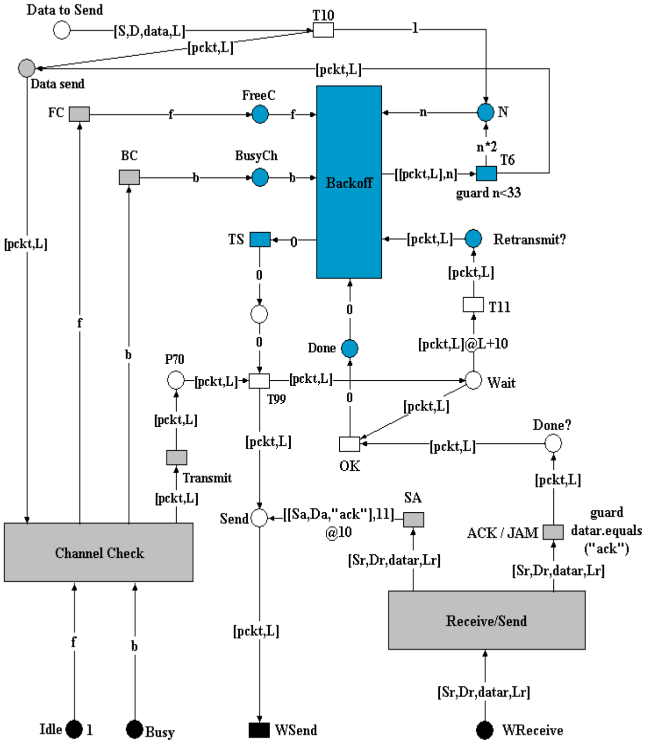


Figure 17. Hierarchical Design of a DCF IEEE 802.11b Workstation Component based on Generic Basic Components

VI. Experimental Validation

In the previous sections, we have modeled several components (basic and composite components). In this section, we will validate and evaluate the quality and accuracy of our model by means of simulation. The obtained results will be compared with the data given by other studies about IEEE 802.11b network and also the results of NS-2 simulations performed in the same conditions.

A. *Performance evaluation techniques*

Different models and methods are used to evaluate the performance in communication and distributed systems [9]:

1) *Measurement* can offer the most exact and accurate results. The system is observed directly. However, measurement is often the most costly of the techniques since it is only possible if the system already exists. In some cases, measurements may not be accurate since it depends on the state of the system. For example, if network measurements are done during peak period, the observed results would not be the same as if the measurements were done during a period of low use of the network.

2) *Analytical* models may provide exact answers. Analytical modeling uses simple mathematical expressions to obtain the performance results for a system. However, these results may not be accurate, because of their dependencies on the made assumptions in order to create the model. The behavior of computer systems including processing periods, communication delays, and interactions over the communication channels is difficult to model. Analytical models are excellent to model small to medium systems that fulfil some requirements but this is not the case for industrial-sized, networked and distributed systems.

3) *Simulation models* [36] allow modeling and observing the system behavior. It facilitates the understanding of the real system. Simulation allows the evaluation of a system model in an executable form, and the data of such process are used to calculate different measures of interest. Simulation is generally used because the complexity of most systems requires the use of simple mathematical ways for practical performance studies. This makes simulation as a tool for evaluation. Simulation models allow creating very detailed, potentially accurate models. However, developing the simulation model may consume a big amount of time, but once the model is built it takes a little time to get results.

Table III shows a qualitative comparison between the different methods used to evaluate the systems performance.

TABLE III. COMPARISON OF THE DIFFERENT PERFORMANCE EVALUATION TECHNIQUES

Criterion	Analytical	Simulation	Measurement
Stage	Any	Any	Post prototype
Time Required	Small	Medium	Varies
Tools	Analysts	Computer Languages	Instrumentation
Accuracy	Low	Moderate	Varies
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High
Scalability	Low	Medium	High
Flexibility	High	High	Low

This comparison is based on different criteria [37] [38]:

- *Stage*: Which performance evaluation technique should be used at any point in life cycle,
- Time required: The time consumed/required by a particular technique,
- *Tools*: Which analytic tools, simulators, measurement packages are used,
- *Accuracy*: It represents the degree to which the obtained results match the reality (evaluates the validity and reliability of the results obtained).
- *Scalability*: It represents the complexity degree to scale a particular technique
- *Trade-off evaluation*: It represents the ability of a technique to study the different system configurations.
- *Cost*: This cost must not be considerable in term of time and money needed to perform the study.
- *Flexibility*: The system-model under test should be easy to modify and extend. The used evaluation technique should provide the possibility to integrate these considerations easily in the developed model.

Simulation seems to be the mostly used technique used to evaluate the performance of the computer systems. It represents a useful means to predict the performances of a system and compare them under many conditions and configurations. One major advantage of this technique is that even if the system is already implemented, it offers flexibility difficult to reach with measurement techniques.

Our modeling formalism, Petri nets, combines both the analytical and simulation models which let the possibility to model system mathematically. However, communication networks and distributed systems are so complex that building and solving the equations' system are too difficult and needs tools capable to perform this process.

More suitable for our aims are the discrete-event simulations [9]. Discrete-event simulation is a powerful computing technique for understanding the behavior of systems. In discrete-event simulations, the state changes occur at discrete steps in time. Discrete event simulation is mainly used in real environments such as communication networks and protocols [39], manufacturing systems [40], material handling [41], etc. General purpose programming languages like C/C++ and Java and several simulators such as NS-2 [42] and OPNET [43] are based on the discrete event simulation.

B. Simulations and Results

To perform the simulations, many tools and extensions of Petri Nets exist such as PROD, Renew, ALPHA/Sim, CPN Tools, Artifex and other tools [44]. However, the development of most of these tools has been stopped for a long time, or they do not support our needs or they are commercial. Two main, free of charge tools were possible to cover the previous features "CPN Tools" [45] and "Renew 2.1.1" [46].

However, during simulation, "CPN Tools" has shown an important problem that does not apply to our timing needs. We have chosen "Renew" since it is a Java-based high-level Petri nets discrete-event simulator.

Our simulations are based on full-mesh dense networks with different numbers of workstations:

i- The simulations were performed for different number of workstations sharing the medium.
ii- For each case, the simulations were repeated 100 times to get average measures.
iii- Each simulation assumes that all nodes transmit at 11Mbps.
iv- All nodes attempt to send data as soon as possible.
v- Each node has 1000 packets (to get the average possible measures) with average packet length of 1150 bytes (packet length varied from 800 byte to 1500byte).
vi- All simulations were accomplished on Intel® Core™ 2 Duo Processor T2300, 2G of RAM.

1) *Average bandwidth per node*

The first result is the average bandwidth per workstation. Fig. 18 shows the throughput of 802.11b nodes sharing a bandwidth of 11Mbps. As illustrated by the figure, the bandwidth per node decreases logically with the increase of nodes number. When the number of nodes is small each workstation has more bandwidth from the shared effective bandwidth. However, when the number of the nodes on the network increases, the bandwidth is decreasing exponentially. This is due to the increased number of collisions on the network, and so more bandwidth will be lost.

The other factor is that CSMA gives fair timing to the machines to access the channel. Thus, workstations must wait longer time to have access to the channel. Another factor is after a collision, the workstations must double their contention window which means longer backoff time. So, more time is spent to decrement the backoff or less total bandwidth.

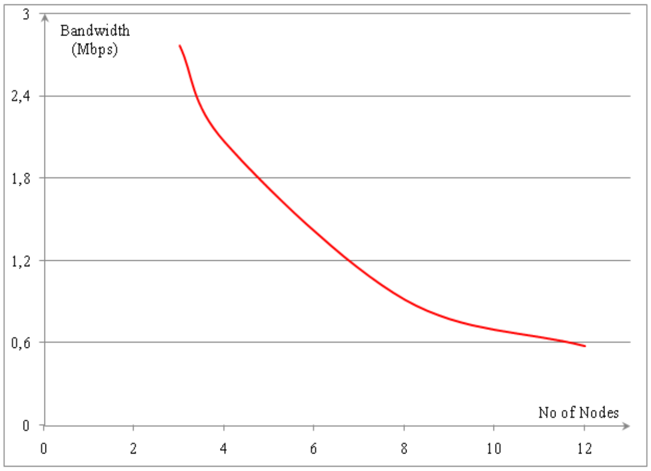


Figure 18. Bandwidth Variation with Number of Nodes

2) *Collisions rate percentage*

The next step is to compute the collision rate percentage or errors versus the network utilization. Fig. 19 shows how the collision rate increases when the number of workstations increases. As we can see in the figure, when three workstations are sharing the medium, the collision rate is nearly 8%. However, when there are 12 workstations sharing the medium, the collision rate reaches 23.2%. These results confirm the results obtained in the previous section and our explanation.

As one can see, the collision rate is increasing linearly until certain point (8 workstations). The reason is when more workstations attempt to send, more packets are on the shared channel and hence the probability that a collision occurs increases. However, when the number increases more, the collision rate increase becomes slower. The explanation for this *evolution* is the *backoff procedure*. With more workstations, the number of collisions increases, and the value of CW also increases (backoff time). On the other hand, this increment of backoff time decreases the probability of a collision, since workstations in collision must wait for longer time before attempting to send again. So, the collision rate increment becomes slower.

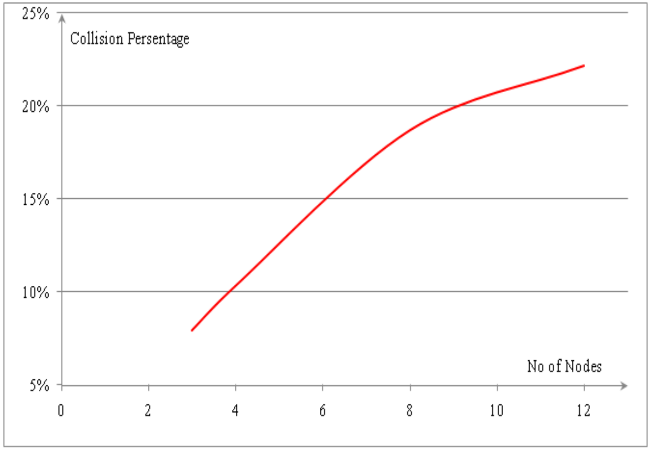


Figure 19. Collisions Rate Percentage

3) *Transmission Time per Packet*

The next test is to measure the overall time needed to send a packet over Ethernet or DCF protocols (from sender side to receiver side). Fig. 20 shows the time required to transmit one packet versus the number of nodes on the network. The transmission time increases linearly due to the increased number of sent packets on the network and collision rate.

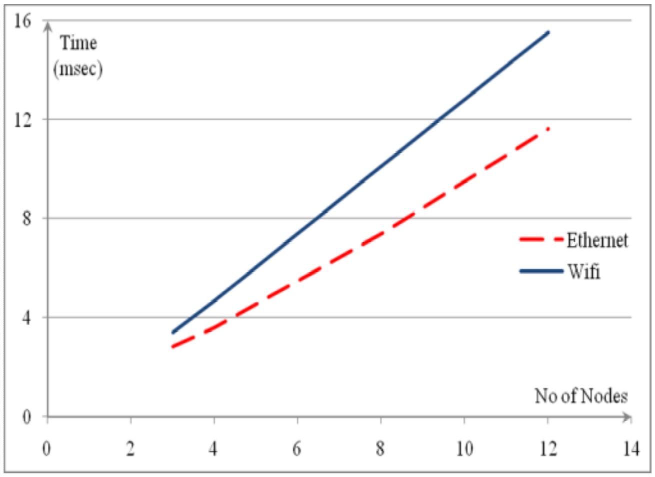


Figure 20. Transmission Time per Packet

However, sending a packet over Ethernet requires less time than sending it over DCF. The figure shows that with three nodes on the network, DCF seems to be the same as Ethernet. However, with the increase of nodes the difference becomes obvious. This is due to:

1- A workstation attempting to use the channel in wireless networks needs to ensure that the channel is idle during a DIFS period or 50µs, while in Ethernet it only needs 9.6 µs.
2- From the first attempt to transmit, wireless nodes starts a backoff procedure (Bavg = 8 * 20 µs) decremented only if the channel is idle, while in Ethernet, workstations defers only for 9.6 µs.
3- After a collision, in wireless networks, the channel status becomes idle only when all the workstations finish their transmissions (no collision detection process), while in Ethernet the channel becomes idle after 51.2 µs (channel acquisition slot time).
4- The backoff procedure used after each collision in wireless networks doubles the contention window value which is already 8 times greater than the one used in Ethernet. This makes the backoff in wireless greater than Ethernet BEB even with slot time (20µs) less than the 51.2 µs used in Ethernet.

C. *Comparison with ns-2 simulator and other studies*

To evaluate the quality and accuracy of our model, we have used the network simulator NS-2 as a comparative tool since it is widely used to model communication protocols. The NS-2 simulator is a discrete-event network simulator that allows simulating many scenarios defined by the user. It is commonly used in the research due to its extensibility, since it is an open source model. NS2 is widely used in the simulation of routing, multicast protocols and ad-hoc network.

Fig. 21 shows the results obtained from NS-2 and those from our model, (Fig. 18). As we can see the results of both simulations Renew and NS-2, are nearly identical which confirms the correctness of our model. Moreover, if we compare our obtained results with those in [Anastasi05] and [Heusse03], we can get also the same results from both the simulation technique and the equation we obtained from the results.

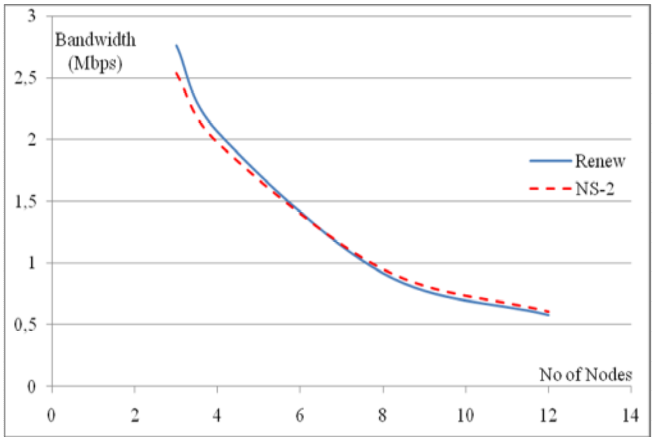


Figure 21. Comparison between our model and NS-2

The other comparison is the effective simulation time. As we can see in Fig. 22, the simulation time increases in a linear way when the number of nodes increases (confirmed by the results in Fig. 20). The figure shows that NS2 needs less time to perform the same simulation. However, NS2 does not support the step-by-step simulation to verify the system event by event. The second important issue is that it is not possible to model distributed services with NS2 (no supporting package). However, with "Renew" as Petri nets editor and simulator, it is possible to combine both services and protocols in one global model.

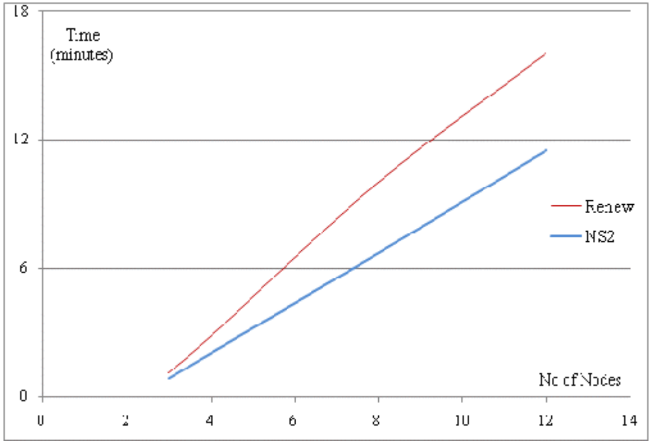


Figure 22. Effective Simulation Time versus number of nodes

VII. CASE STUDY: EVALUATING PERFORMANCE OF A DISTRIBUTED MANUFACTURING SYSTEM

In the last sections, we have shown the modeling part of the communication protocols. In this section we will show the modeling part that concerns the services. An illustrative example, Fig. 23, will be used to model the services offered by a production system. The used modeling technique will be the same as the communication protocols, i.e. component-based methodology, where each part of the system is modeled in hierarchical composition: "*service-workstation*", i.e. each service is modeled over a workstation.

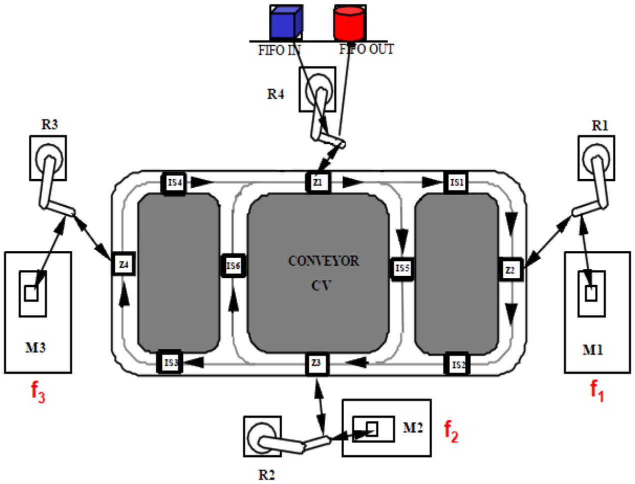


Figure 23. Manufacturing Plant with Flexibilities

A. *System Components*

Fig. 24 shows the complete system components used to transfer one product from *IN/OUT* area to any machine *M1*, *M2* or *M3*. *Z1* to *Z4* represent the input and output areas for each machine and the IN/OUT area. The capacity of each is limited to one product. *IS1* to *IS6* areas represent the stock area before and after machining a product. The capacity of each stock area is greater than one. R1 to R4 represent the robots used to make a transfer from a Z area to a machine or IN/OUT area and vice versa. Finally, *T1* to *T8* represent the transfer elements from and to a Z and an IS stock areas. The component represents the service offered by a resource, a robot or a transfer component. Machines, Z's and stock areas are considered as shared resources.

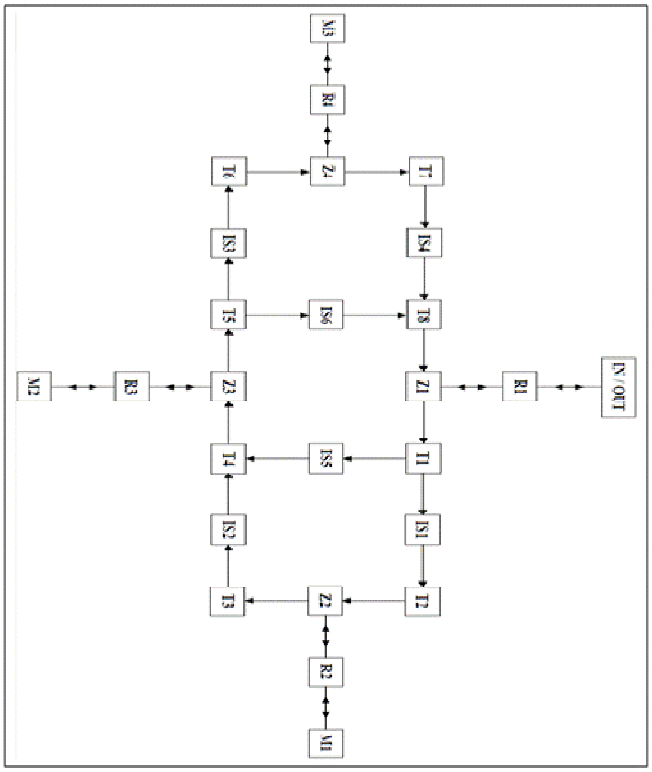


Figure 24. Complete Area and Transfer Components

B. *Product transfer*

In order to transfer a product from one area to another one, areas must allocate the required area/resource. Pre-allocation is passed through a transfer component. Transfer components check the possibility to allocate the destination area (depending on the capacity of each area). An acknowledgement is from the destination area when a place is free. During this time the source area and the transfer component are in waiting period (machines and Z areas do not perform any action during this time, while stock areas can receive products from other components).

Fig. 25 shows the centralized model of a product transfer from S to D areas. In the figure, to transfer a product, the product must be available in area S (a token put in place S/REQ), the transfer component must be also available (a token in place S→D/NOP) and a free place in area D (a token in place D/CONS). These three tokens enable the transition S→D/t1 and a token is then put in place S→D/TRSF-START starting the transfer process. The transfer component takes the product from area S. The firing of transition S→D/t2 and the put of a token in place S/ACK inform that a place is released up in area S. The transfer process continues by putting a token in place S→D/TRSF-END. When the product arrives to area D (transition S→D/t3), the transfer component becomes free again (a token is put in place S→D/NOP) and an area is used in area D (a token is put in place D/PROD).

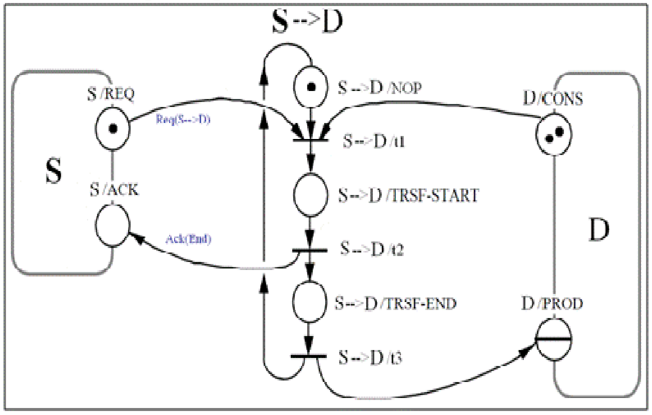


Figure 25. Product Transfer in centralized model

Fig. 26 shows the complete messages exchanged in case of implementation of the 3 processes (S, S→D, and D) on 3 different computers.

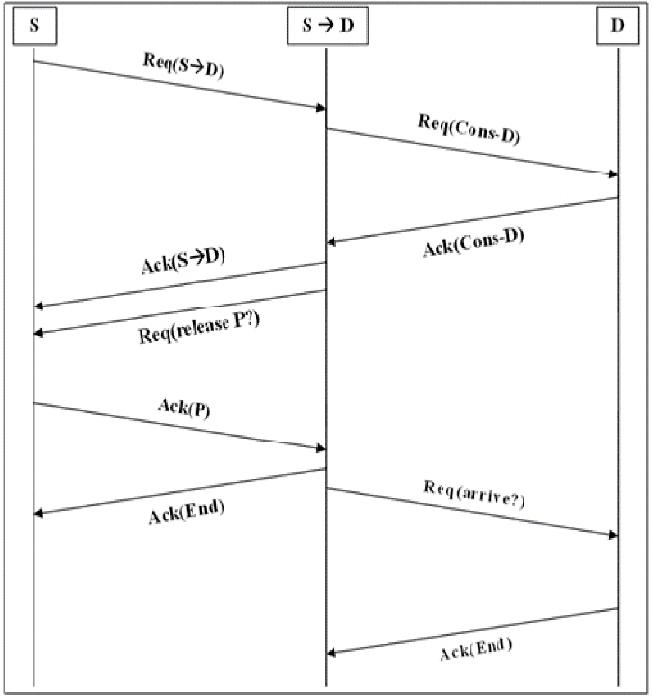


Figure 26. Exchanged Messages over the Network for the Transfer

Each process plays a different role with regard to the client/server mechanism. S is always a client and D is always a server. The role of S→D varies depending on the message. At first, the source area *S* (workstation) sends a request message to the transfer workstation, *S→D* (T_i or R_i), containing the destination workstation *D*. T_i (or R_i) sends a request to D, requesting a free place (*Cons-D*). If there is a free place, *D* will send a positive acknowledgment to T_i (or R_i), otherwise S and T_i (or R_i) will stay in a waiting period. Once T_i (or R_i) receives the acknowledgment, it sends two messages to S containing a positive acknowledgment and a request to release the product. When the product is released S sends an acknowledgment to T_i (or R_i) to start the transfer. When T_i (or R_i) takes the product, it sends an end message to S to free one its places (Cons-S). Finally, it sends a message to D asking the arrival of the product to its side. Once the product arrives to D, it sends an acknowledgement to T_i (or R_i) informing the end of the transfer.

C. Modeling the service components

Fig. 27 shows the complete messages sent and received by a transfer component. The component receives a request packet from an area (transition t60). In order to validate this request a token must be present in place “Cons”, representing the capacity of this component. It sends a message to the destination area requesting a free place. The component stays in a waiting period (place p60) until it receives acknowledgment packet from the destination area (transition t62).

Once the acknowledgment arrives, the transfer component sends request (“release P?”) to the source area. To insure a proper functionality of this module, a guard is associated to the transition t63 to assure that the sender of the acknowledgment is the destination area. Again, the component stays another time in waiting period (place p62) until it receives the second acknowledgment.

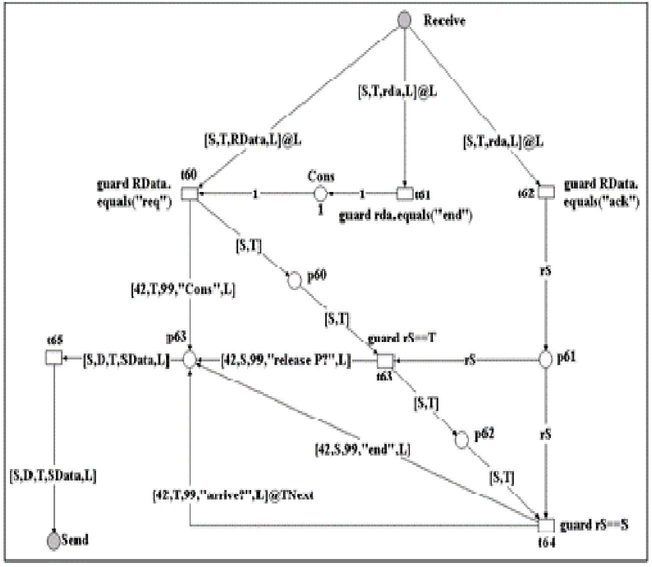


Figure 27. Transfer Component – Service Part

When the acknowledgment arrives, transition t64 is enabled (condition: the sender must be the source), two messages are sent: to the sender releasing one place (the product is taken by the transfer component), and to the destination area requiring if the product has arrived. This second message is sent when the first message is sent (the *TNext* inscription on the arc between t64 and p63).

Fig. 28 shows a complete Petri Net model for Z1 area. In the figures, when a product arrives to the area, a procedure of exchanged messages starts depending on the destination area until that product is transferred to its final destination. Hence, only the workstation component is connected to the network, while the output interfaces of the service component (dark gray) is connected directly to their destination components.

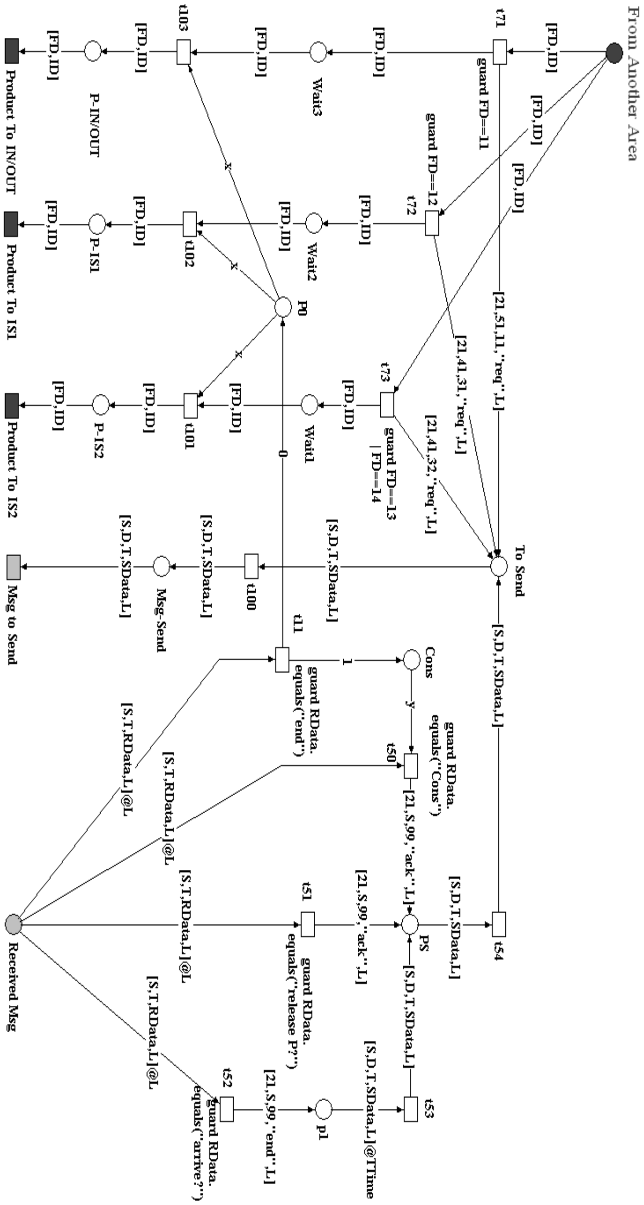


Figure 28. Hierarchical Service-Workstation Petri Net

Fig. 29 shows a sub-model for a transfer of a product from area Z1 to stock IS1 through the transfer element T1. In the figure, the modeled components are reused to build the whole system. This reuse is applicable for the other elements in the system. Some additional places and transitions are used to complete the system-model and to insure its functionality.

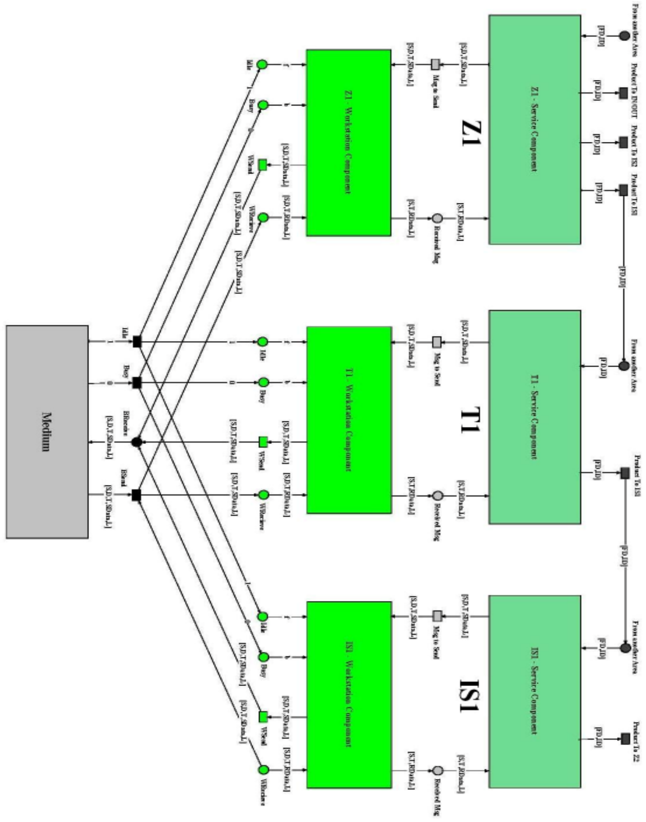


Figure 29. Service-Workstation Composite-Component

D. *Simulating the complete model*

The simulation was performed on the same PCs used in the above sections. The system is assumed to perform 100 different products. The simulation aims to see the impact of using different type of products and different protocols over the system. The transfer time is supposed to be 50 msec and the machining time to be 100 msec. These values have been chosen in milliseconds to really verify the impact of the underlying network on the system. Otherwise, if we use the real values in minutes, the impact of the underlying network would not be obvious with the example we have used. The number of simultaneous products per type is varied from 2 to 5 products.

The type of services on the system affects the number of exchanged messages and transactions on the network. For example, to perform the service f2, the number of transactions is 72 exchanged messages per product. However, to complete service f1 or f2, the number of exchanged messages is 90 messages per product. This is in the case of one product only on the system. However, when there are several products on the system, this number increases due to collisions. So, this number may reach 90~100 messages per product for service f2, and 110~120 messages per product for service f1 or f3.

a) One Product

The first simulation is performed to get an idea about the time needed to machine one product over the system. Table IV shows the impact of changing the communication protocol in the system over the time needed to finish one product. An important difference appears between Ethernet at 10Mbps and 100Mbps. However, the 1Gbps does not create a big difference, since the machining and transfer times are the dominant in this case.

TABLE IV. TIME TO MACHINE A PRODUCT

Service	802.11b	E-10Mbps	E-100Mbps	E-1Gbps
f2	564.5 ms	567.6 ms	506.7 ms	500.7 ms
f1 or f3	680.2 ms	684.5 ms	608.5 ms	600.9 ms

The other interesting result is the time difference when the required service is f2, or f1 or f3. Since the path to finish the product is longer, the time needed to make the product is clearly longer. In this part, 11Mbps 802.11b seems to be better than 10Mbps Ethernet.

b) Same Products; Different Protocols

The second results are the most important, since they show the impact of changing the communication protocol on the system. Different remarks can be done from the Fig. 30:

1- 802.11b protocol does not present a good choice. This result is conforming with the results of Fig. 20. This becomes clear when the number of simultaneous products increases (the number of exchanged messages increase also).

2- A big time difference is noticed when using 100Mbps Ethernet (compared to 10Mbps Ethernet and 802.11b). The number of messages is important. With 2 simultaneous products of each type, the number of exchanged messages reaches 500 to 600 exchanged messages. With 3 simultaneous products of each type, the number of exchanged messages reaches 900 to 1000 exchanged messages. While with 5 simultaneous products of each type, there are nearly 1400 to 1500 exchanged messages on the network.
The type and speed of protocols is very important since to exchange this huge number of messages on the network, the bit rate is very important and decreases obviously the time needed to exchange these messages between the different resource/workstation on the system.

3- The use of 1Gbps Ethernet did not show a big difference with respect to 100Mbps Ethernet. However, this conclusion is not really correct. The impact of using Giga Ethernet can appear if the modeled system is larger (more machines, stock areas, resources, etc.).
In that case, the number of exchanged messages over the network will be greater. Thus, the impact

of using Giga Ethernet will become obvious since the time needed to send these messages will be shorter (for example, as the time difference between 10 and 100Mbps).

However, in our model the number of modeled components is still medium (3 machines, 4 resource areas and 6 stock areas). So, the machining and transfer times are dominant here when using Giga Ethernet compared to 100Mbps Ethernet.

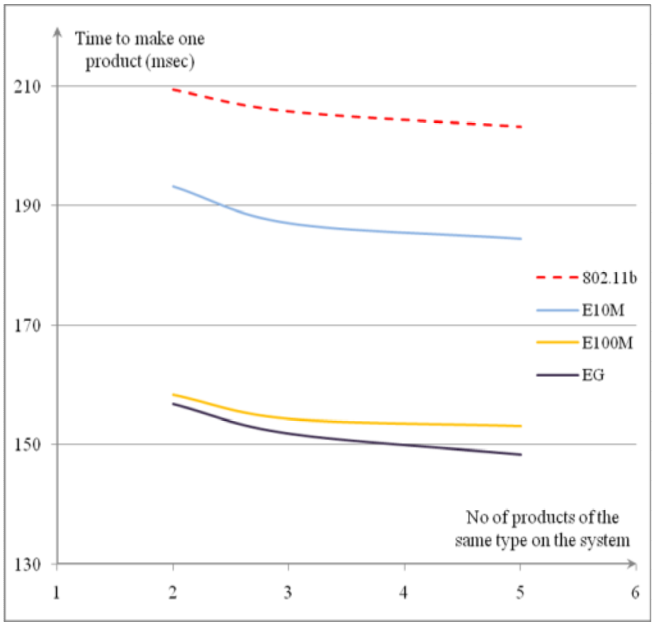


Figure 30. Impact of changing the communication protocol in the system

VIII. Conclusion

Distributed systems are more and more present in our daily life. These systems are complex and can be in one place or even everywhere in the world. The use of distributed systems allows sharing different and expensive resources by a several clients. Thus, the need to control the distributed systems is very important. Manufacturing systems are one kind of these systems.

The need to model these systems before their implementation is important. The design stage allows verifying some of their properties. A well-designed model that takes into accounts all the requirements and constraints of a system can save cost and time.

In this work, we have presented the problem of modeling manufacturing systems and the underlying communication protocols. However, modeling a huge and complex system implies to have also a big and complex model. So, we have proposed in this work a component-based modeling approach based on High-Level Petri Nets.

This approach can meet the challenges of modeling the distributed systems and the communication networks. Genericity, modularity and reusability are the main and important characteristics of this approach since it allows reusing ready-to-use components and easily fitting them to new system-models depending on the requirements of clients and applications. These advantages allow building complex models in an easier way.

References

[1] A. Tanenbaum, “*Distributed Operating Systems*”. Prentice Hall, 1995.

[2] G. Coulouris, J. Dollimore, and T. Kindberg, “*Distributed Systems: Concepts and Design*”, 3rd ed. Pearson Education, 2001.

[3] A. Toguyeni, “*Design of Modular and Hierarchical Controllers for Reconfigurable Manufacturing Systems*.” IMACS Multiconference on Computational Engineering in Systems Applications, Vol. 1, pp. 1004-1011, 2006.

[4] H. Sarjoughian, W. Wang, K. Kempf, and H. Mittelmann, “*Hybrid discrete event simulation with model predictive control for semiconductor supply-chain manufacturing*.” Proceedings of the 37th Conference on Winter Simulation, pp. 256 – 266, 2005.

[5] P. Berruet, J. Lallican, A. Rossi, and J-L. Philippe, “*A component based* approach for the design of FMS control and supervision.” IEEE International Conference on Systems, Man and Cybernetics, Vol. 4, pp. 3005-3011, 2005.

[6] A. Masri, T. Bourdeaud'huy, and A. Toguyeni, “*Performance Analysis of IEEE 802.11b Wireless Networks with Object Oriented Petri Nets*”, Electronic Notes in Theoretical Computer Science, Proceedings of First International Workshop on Formal Methods for Wireless Systems FMWS’08/CONCUR’08, Vol. 242, No 2, pp. 73-85, Canada, 2008.

[7] A. Masri, T. Bourdeaud'huy, and A. Toguyeni, “*Network Protocol Modeling: A Time Petri Net Modular Approach*.” 16th International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2008, pp. 274-278, Croatia, 2008.

[8] A. Masri, T. Bourdeaud'huy, and A. Toguyeni, “*A Component Modular Modeling Approach Based on Object Oriented Petri Nets for the Performance Analysis of Distributed Discrete Event Systems*,” Fifth International Conference on Networking and Services ICNS, pp.222-227, Spain, 2009.

[9] C. Cassandras and S. Lafortune, “*Introduction to Discrete Event Systems*”, 2nd ed. Springer Science+Business Media, LLC, 2008.

[10] P. Ramadge and W. Wonham, “*The Control of Discrete Event Systems*.” Proceedings of the IEEE, Vol. 77, No. 1, 1989.

[11] P. Paraskevopoulos, “*Modern Control Engineering*”. Marcel Dekker, Inc., 2002.

[12] R. Burns, “*Advanced Control Engineering*”. Butterworth-Heinemann, 2001.

[13] W. Zhang, M. Branicky, and S. Phillips, “*Stability of Networked Control Systems*.” IEEE Control Systems Magazine, Vol. 21, pp. 84-99, 2001.

[14] F. Wang and D. Liu, “*Networked Control Systems: Theory and Applications*”. Springer-Verlag London Limited, 2008.

[15] N. Mir, “*Computer and Communication Networks*”. Prentice Hall, Inc, 2007.

[16] W. Stallings, “*Data and Computer Communications*”, 8th ed. Prentice-Hall, Inc, 2007.

[17] H. Zimmermann, “OS1 Reference Model-The IS0 Model of Architecture for Open Systems Interconnection.” IEEE Transactions on Communications, Vol. COM-28, No. 4, 1980.

[18] Unified modeling language UML : *http://www.omg.org/uml* - 2009.

[19] J. Peterson, “*Petri Net Theory and the Modeling of Systems*”. Prentice-hall International, 1981.

[20] P. Ramadge and W. Wonham, “*Supervisory control of a class of discrete event processes*.” SIAM Journal on Control and Optimization, Vol. 25, No. 1, pp. 206-230, 1987.

[21] F. Lin and W. Wonham, "*Decentralized supervisory control of discrete event systems with partial observation*." IEEE transaction in Automatic Control, Vol. 35, pp. 1330-1337, 1990.

[22] E. Zamaï, A. Chaillet-Subias and M. Combacau, "*An architecture for control and monitoring of discrete events systems*." Computers in Industry, Vol. 36, No. 1 - 2, pp. 95-100, 1998.

[23] J. Petin, P. Berruet, A. Toguyeni, and E. Zamaï, "*Impact of information and communication emerging technologies in automation engineering : outline of the intica proj*." 1st Workshop on Networked Control System and Fault Tolerant Control, 2005.

[24] C. A. Petri, "*Communication with Automata*." Technical Report RADC-TR-65-377 Rome Air Dev. Center, New York, 1966.

[25] K. Jensen, "*Coloured Petri nets: A high level language for system design and analysis*", Lecture Notes in Computer Science, Vol. 483/1991, pp. 342-416, 1991.

[26] P. Brereton and D. Budgen, "*Component-Based Systems: A Classification of Issues*." IEEE Computer, Vol. 33, No. 11, pp. 54-62, 2000.

[27] D. Carney and F. Long, "What Do You Mean by COTS? Finally, a Useful Answer." IEEE Software, 2000.

[28] G. Gössler, S. Graf, M. Majster-Cederbaum, M. Martens, and J. Sifakis, "*An Approach to Modelling and Verification of Component Based Systems*." Lecture Notes in Computer Science, SOFSEM 2007: Theory and Practice of Computer Science, Vol. 4362, pp. 295-308, 2007.

[29] R. Bastide and E. Barboni, "*Component-Based Behavioural Modelling with High-Level Petri Nets*." In MOCA'04 Aahrus, Denmark, DAIMI, pp. 37-46, 2004.

[30] R. Langlois, "*Modularity in technology and organization*." Journal of Economic Behavior & Organization, Vol. 49, p. 19–37, 2002.

[31] C. Geisterfer and S. Ghosh, "*Software Component Specification: A Study in Perspective of Component Selection and Reuse*." Fifth International Conference on Commercial-off-the-Shelf (COTS)-Based Software Systems, pp. 100-108, 2006.

[32] A. Brown and K. Wdlnau, "*Engineering of Component-Based Systems*." Second IEEE International Conference on Engineering of Complex Computer Systems, 1996.

[33] IEEE 802.3 Ethernet Working Group: *http://www.ieee802.org/3/* - 2009.

[34] IEEE Computer Society, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications." IEEE Std. 802.11™, 2007.

[35] IEEE Std 802.3™, "Carrier sense multiple access with collision detection (CSMA/CD) access method and physical layer specifications." 2002.

[36] P. Fortier and H. Michel, "Computer Systems Performance Evaluation and Prediction". Digital Press , 2003.

[37] A. Chhabra and G. Singh, "*Parametric Identification for Comparing Performance Evaluation Techniques in Parallel Systems*." Proceedings the 1st National Conference on Challenges and Opportunities in Information Technology, COIT, pp. 92-97, 2007.

[38] R. Jain, "The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling ". John Wiley & Sons,Inc., 1991.

[39] H. Lim, B. Wang, C. Fu, A. Phull, and D. Ma, "*A Middleware Services Simulation Platform forWireless Sensor Networks*." The 28th International Conference on Distributed Computing Systems Workshops, ICDCS, pp. 168-173, 2008.

[40] N. Kumar and R. Sridharan, "Simulation modelling and analysis of part and tool flow control decisions in a flexible manufacturing system ." Robotics and Computer-Integrated Manufacturing, 2009.

[41] V. Giordano, J. Zhang, D. Naso, M. Wong, F. Lewis, and A. Carbotti, "*Matrix-based discrete event control of automated material handling systems*." Proceedings of the 45th IEEE Conference on Decision & Control, 2006.

[42] NS2 Official Website:

http://nsnam.isi.edu/nsnam/index.php/Main_Page - 2008.

[43] OPNET Technologies, Inc. *http://www.opnet.com/* - 2009.

[44] Petri Nets World: http://www.petrinetz.de/ - 2009.

[45] Computer Tool for Coloured Petri Nets . CPN Tools: *http://wiki.daimi.au.dk/cpntools/cpntools.wiki* - 2007.

[46] The Reference Net Workshop. Renew : *http://www.renew.de/* - 2008.

Improving P2P Streaming Methods for IPTV

Majed Alhaisoni
University of Essex
Colchester, UK
malhai@essex.ac.uk

Antonio Liotta
Eindhoven University of Technology
The Netherlands
A.Liotta@tue.nl

Mohammed Ghanbari
University of Essex
Colchester, UK
ghan@essex.ac.uk

Abstract

Peer-to-Peer (P2P) IPTV applications have increasingly been considered as a potential approach to online broadcasting. These overcome fundamental client-server issues and introduce new, self-management features that help improving performance. Recently, many applications such as PPlive, PPStream, Sopcast, and Joost have been deployed to deliver live and Video-on-Demand streaming via P2P. However, the P2P approach has also shown some points of failure and limitations. In this paper we analyze, assess and compare two popular Live and Video-on-Demand P2P streaming applications, Sopcast and Joost. Fundamental shortcomings of existing applications are then countered by our approach where we employ a cross-layered method aimed at improving network efficiency and quality of experience. We show how simple modifications to existing protocols have the potential to lead to significant benefits in terms of latency, jitter, throughput, packet loss, and PSNR.

Keywords: P2P streaming; Ne; Network efficiency; Load balancing; QoS; QoE.

1. Introduction

Peer-to-Peer (P2P) video streaming is becoming a viable alternative to conventional IPTV systems for distributing video content over the Internet. The underlying mechanism is based on the distribution of the stream through a self-managed, application-level overlay including the user terminals in the role of peers i.e., content distribution relays. This is in contrast to other IPTV approaches which are based on content distribution networks. These require a dedicated multicasting infrastructure whose cost increases dramatically with the scale and dynamics of the system. On the other hand, in the P2P approach any capable (user) terminal becomes a distribution hub for any incoming stream, reducing in this way the possibility of failure points and bottlenecks that are traditionally associated with servers. Moreover, as the number of connected users increases, the number of distribution points grows too. As a consequence, the system scales much better than any client-server counterpart.

The P2P streaming concept has nowadays been deployed into several trial P2P streaming systems, such as Sopcast [2], Joost [3], and Zattoo [4]. The online broadcasting arena is then evolving, mainly due to the clear commercial interest for these new technologies. Many hosts can be supported by a P2P multimedia system, possibly in excess of hundreds or even millions, with miscellaneous heterogeneity in bandwidth, capability, storage, network and mobility. However, although most of the P2P applications are designed to automatically load-balance computing resources, they fail to pursue network efficiency.

Accordingly, this article begins by describing experiments to assess the network efficiency in two of the most popular P2P applications that is Sopcast for the real-time broadcasting and Joost for the Video-on-Demand (VoD) services. By analyzing traffic traces we calculate and compare the network efficiency of these systems, in terms of network locality and percentage of P2P traffic (as a fraction of CS (Client-Server) traffic). The percentage of P2P traffic, gives another form of computational efficiency since the aim of P2P systems is to minimize server intervention. We find, however, that due to the particular nature of P2P streaming, it is not always possible to do without server support and that various systems address this issue in radically different ways. P2P traffic percentage gives an indication of both computational and network load balancing. We also look at the latter property from the view point of network locality, which is the ability to keep traffic local (in addition to being spread out). Our analysis reveals that network efficiency is being exploited poorly in the conventional P2P streaming systems, which indicate new opportunities for designing future applications aiming at network computational and cost efficiency.

Following our analysis of two representative P2P platforms, we then propose ways in which simple modifications to existing protocols have the potential to lead to significant benefits in terms of latency, jitter, throughput, packet loss, and PSNR (Peak signal to noise ratio). The important aspect of locality shows up in our results as it confirms that user satisfaction and network locality cannot be treated independently. For instance in order to satisfy more users, network resources should be treated efficiently which will in return help in supporting more users. Our method aims at improving network utilization and locality whilst at the same time not degrading computational load balancing. The proposed scheme is run under the ns-2 simulator [27].

2. Related Work

Many studies have been published about P2P streaming, but very few actually focus on the analysis of the inter-relationship between computational and network efficiency deriving from self-managed, P2P networks. Existing work aims at understanding the underlying algorithms of current applications which are mostly proprietary. Single-systems analysis has been carried out, like for example in the case of Joost. Consideration on network utilization and locality are given in [5] and [6].

Other studies are based on the comparison of two or more P2P applications for video streaming [7]. However the focus is on examining similarities and differences from the structural point of view, looking at protocol utilization, percentage of download and upload traffic, and signaling overheads. This is the case, for example, of the work published by Silverston and Fourmaux about Sopcast, TVAnts, and PPlive [7], and by Ali et al. about Sopcast and PPlive [8] [9]. Other studies have considered different applications such as [10] and [11]. Moreover, Gustavo Marfia et al [12] have conducted an experimental study on Video-On-Demand system and they were concerned about the performance of the system on different environments such as campus and residential. Thus, they gave some results about protocol utilization and the start-up delay. Additionally, they gave brief descriptions for the existing architectures such as Tree and Mesh.

On the other hand, ways to pursue efficiency between overlay and underlay networks have started to be investigated only recently. Authors in [13] propose a technique, where the peers on the overlay are chosen based on their mutual physical proximity, in order to keep traffic as localized as possible. A similar approach is described in [14], where they measure the latency distance between the nodes and appropriate Internet servers called landmarks. A rough estimation of awareness among nodes is obtained to cluster them altogether, as in [15] [16].

Overlay locality is studied also by [17], where the authors make use of network-layer information (e.g. low latency, low number of hops and high bandwidth). We use though a different distance metric, based on RTT (round trip time) estimations, to prioritize overlay transmissions. Additionally, we use a cluster management algorithm whereby communicating peers are forced to periodically handover, in order to pursue computational as well as network efficiency (as explained in [1] and [23]).

Hefeeda et al [18] have proposed a mechanism for P2P media streaming using Collectcast. Their work was based on downloading from different peers. They compare topology-aware and end-to-end selection based approaches.

The latter approach is also the subject of [19], which employs a simpler version of our RTT approach based on continuous pinging of peers. Similarly, we adopt clustering to limit the signaling overheads associated with this process and prevent bottlenecks.

Other studies such as [20], propose relevant methods to serve multiple clients based on utility functions or clustering. A dynamic overlay capable of operating over large physical networks is presented in [21]. In particular, they show how to maximize the throughput in divisible load applications.

Looking at previous studies, we can say that our main contributions are:

1. To complement other on-going studies with a view to better understand the impact of P2P streaming onto the network. Our main attention is on network efficiency and locality, with the aim to identify intrinsic deficiencies of existing platforms.

2. To propose a new approach to improve network locality.

3. To study a new combination of existing techniques (cross-layer optimization, localization, switching over, forced handovers).

4. To take the perspective of the network operator, in trying to harmonize overlay and underlay networks

5. To quantify the impact of the network-aware approach on QoS (Quality of Service) and QoE (Quality of Experience) factors.

3. Experimental Setup

Our experiments were conducted in the United Kingdom. Measurements and data collection have been carried out during Euro 2008, a major event when thousands of users adopted P2P Sopcast application to watch football matches live. We collected a unique data set, measuring most of the Euro games. We have collected 1.10 GB and that was observing the whole match which is around 1 hour and 30 minutes.

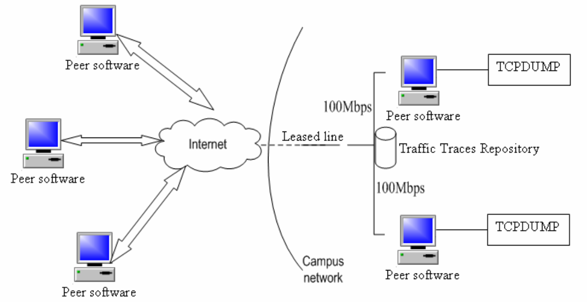


Figure 1 - Experimental Environment

On the other hand, VoD data was collected from P2P application Joost version 1.1.4. The overall size of our trace files was 277 Mbytes. Moreover, it is worth mentioning that the collected data for both live and on demand includes the video of the game which was sent to our client in UK. However, Figure 1, outlines the set up of the two machines used for the collection of traces, which were then used for packet analysis. The machines were connected to a 100Mbps Ethernet, connected to the campus Internet leased line. This ensured that both inbound and outbound bandwidth were considerably higher than the minimum required for the correct functioning of Joost and Sopcast which is 300-350 kbps for Sopcast and 500 kbps for Joost [5].

4. Performance Metrics

Every single application can make use of different strategies in order to provide a video streaming service. However, there are some features that a P2P streaming system should have in order to fully benefit from resource distribution and load balancing. First of all, for efficient network utilization, locality is one of the main goals. Network locality is the ability to maintain the P2P overlay in such a way as to create logical connections among peers who are physically close to each other. The ideal condition occurs when the most intensive data exchanges happen among nearby peers.

4.1. Network Efficiency

In order to evaluate the network efficiency of the two scrutinized applications, we first had to come up with a way to measure it, taking into consideration factors that have an influence. Our approach has been to weight positively those characteristics that improve efficiency and *vice versa*.

We considered 15 observation windows, each of one minute duration. Within each window we considered the relative distance among all pairs of peers in terms of average Round Trip Time (RTT) and its relation with the amount of exchanged traffic volume (expressed in Mbytes). The result is weighted by the minimum RTT obtained during the whole observation period (15 minutes) divided by the total of traffic expressed in Mbytes. Moreover, it is commonly agreed that the lower is the "RTT", the more would be the offered data rate (bandwidth). Thus, peers offering lower RTTs are considered as having a higher available bandwidth and that may be shared fairly among the services.

$$Ne = \sum_{i=1}^{n} \left(\frac{t_i}{\sum_{j=1}^{n} t_j} * \frac{\min(d_i)}{d_i} \right) \qquad \text{[Eq1]}$$

We have come up with Eq 1 which gives an account of network efficiency.

Where Ne is the network efficiency and i = 1 …. n identifies the set of connected hosts; t_i is the traffic coming into the node under scrutiny (or client node) from each peer *i*, (expressed in Mbytes); $\sum t_j$ represents the total traffic coming from all the connected hosts (considering all the observed peers in a window); d_i represents the distance between the intending client and each peer *i*, expressed as Round Trip Time (RTT), min (d_i) indicates the minimum distance between the client and all other peers across the observed window. Hence, we can define the efficiency factor for each of peer in a minute window.

4.2. Peers vs. Servers

In order to ensure uninterrupted streaming experience, P2P applications make use of servers that kick in the streaming process when P2P traffic cannot meet the necessary delivery deadlines. Hence, another measure of efficiency is the ratio between client-server (CS) and P2P traffic incurred in the network. Clearly, an efficient system is the one in which most of the traffic comes from peers and, particularly, when peers are physically sitting close to each other.

Unfortunately, we could obtain an exact measurement of this CS to P2P ratio only in the case of Joost, as Sopcast seems to rely on foreign companies (or websites) that supply CS content from inaccessible locations. Nevertheless Sopcast makes also use of a large number of back-end servers in order to increase quality and availability.

5. Results

In this section, we show our results categorized by type of streaming models, which is Real Time and Video-on-Demand. We look at two difference scenarios: a peak-demand scenario exemplified by the Euro 2008 event, and normal operational model.

5.1. Real Time

We have based our observations on Sopcast, which is currently a popular real-time P2P streaming application.

5.1.1. First Scenario (peak demand)

The Euro 2008 event gave us a unique opportunity to assess the system behavior, particularly network locality, when a large number of users connected globally.

Given that our test-bed was in the United Kingdom, we wanted to make sure that a large user population was based in Europe to be able to collect a representative data set. Hence we recorded traffic traces during the match between Czech Republic and Switzerland. It's noticeable from figure 2 that the network locality (as measured with Eq. 1) starts at low values but gradually increases as it manages to prioritize connections based on the mutual distance of peers.

This positive trend is, however, reversed when the number of peers starts increasing more rapidly, that is after about the first 7 minutes of the match. Figure 3 shows a significant increment of peers (*Note that the total number of peers of figures 3, 5 and 7 relate to the peers that connect to our UK test-bed. This number is much smaller than the total number of peers connecting to the P2P system but it gives an indication of the trend of peer number and churn*).
after 7 minutes which is followed by a sudden decrease in network efficiency (Figure 2).

When we looked a bit more closely at what happened during this sharp fall, we realized that not only peers were increasing but there was also a significant churn of peers *(peers leave and join the session regularly)*, that is many peers where disconnecting while even a greater number was actually joining the video stream. Between minutes 8 and 12 the churn of peers was still quite considerable and Sopcast did not manage to improve on network efficiency. A fall in connected users between minutes 12 and 13 was followed but a sharp improvement in efficiency but this was reversed by a sharp increase of users at min 14.

These results tell us that Sopcast manages to pursue network efficiency even in the large scale, but only if churn is limited.

5.1.2. Second Scenario (normal operation)

The second scenario that we studied was during an ordinary show. The most apparent result from Figure 4 is that, apart from the initial start-up period, the system manages to maintain a more stable level of network efficiency.

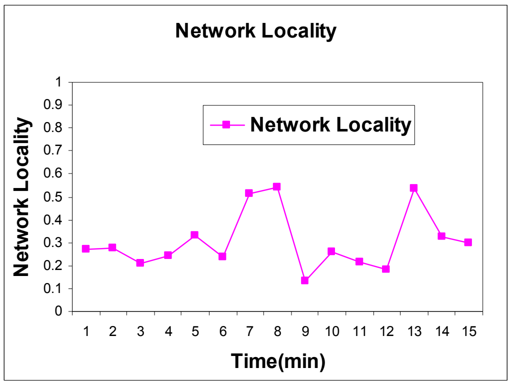


Figure 2 - Network Locality (Sopcast) (Euro 2008)

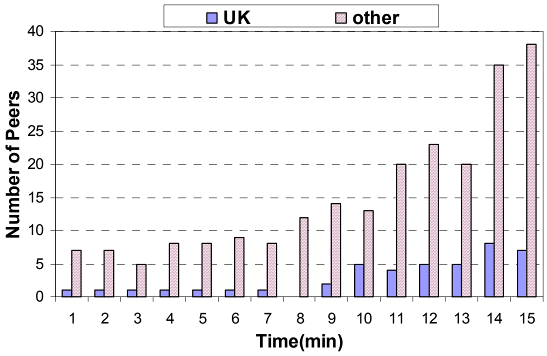


Figure 3 - Number of Peers (Euro2008)

This is because, at steady state (i.e. once people have decided that they will watch the show) the level of churn is limited and Sopcast can manage the overlay more efficiently. Looking more closely at what happens during the first 4 minutes; we can notice that this is the time users are making a decision as to whether they will watch the show. Figure 5 indicates a sharp decrease in watchers. Our in-depth observations also showed that there was also significant churn, hence the poor network performance.

We also notice that, although most of the traffic is coming from outside the UK, there is a good level of locality. This was due to the fact that the network was relatively uncongested with relatively low values of RTTs.

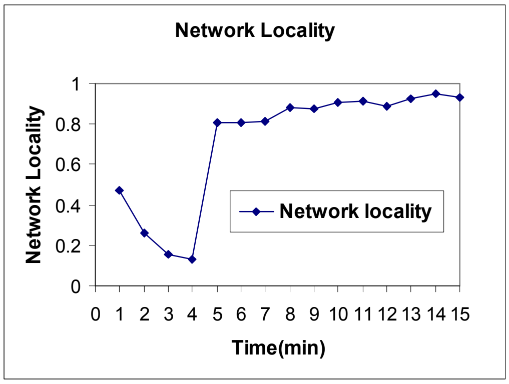


Figure 4 - Network Locality (Sopcast) (Ordinary Day)

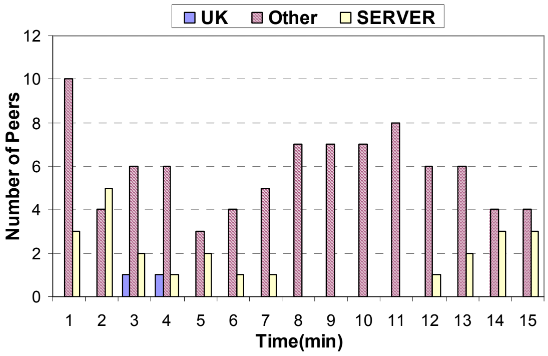


Figure 5 - Number of peers (Ordinary)

5.2. Video-On-Demand

Video-On-Demand traces were taken from Joost which has emerged more recently but is already gaining significant attention. Joost operates in a rather different way than Sopcast [5] [6]. It employs a statistical load-balancing algorithm which seems more efficient than other systems from the computational viewpoint [5]. Our study shows, however, that this benefit is achieved at the expenses of network efficiency. Our experiments were conducted running Joost in our UK-based test-bed. Traces relating to the most popular Joost channels were collected. Equation 1 was used to benchmark network efficiency as for the case of Sopcast.

By looking at the combined results from Figures 6 and 7 we can draw some lessons about the approach followed by Joost. The most apparent difference with the results obtained from Sopcast is that in Joost there is a continuous fluctuation in network efficiency. This results from the fact that Joost continuously forces handover among the intercommunicating peers, with the aim to maximize computational load-balancing.

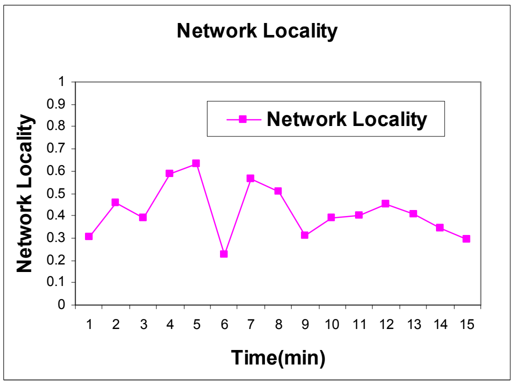


Figure 6 - Network Locality (Joost)

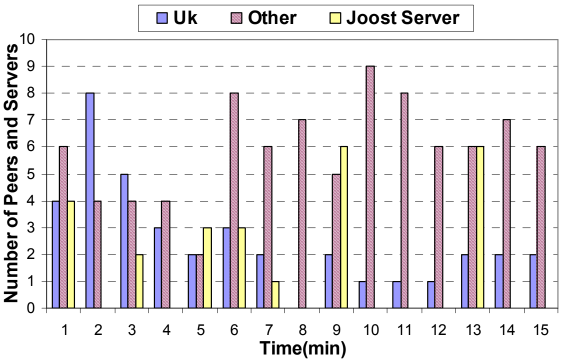


Figure 7 - Number of peers (Joost)

Because inter-communicating peers are forced to handover, network efficiency is dominated by varying link conditions (i.e. RTT values).

To counter the best-effort nature of those links, Joost deploys streaming servers in strategic locations. Currently there appear to be 2 servers in the US and 2 in Europe (one of which in the UK). Each server has many clients to upload the content to the receivers and this is clear from figures 6 and 7 where we sometime notice number of servers up to 5.

When those servers kick in, network efficiency drops, as is visible for instance at minutes 6 and 9. Figure 7 actually shows that the server-originating content is considerable in Joost. Also the amount of P2P traffic coming from outside the UK is significantly higher than local traffic. Again, this indicates the network-unfriendly behaviour of Joost. Hence, even though the user-perceived quality is good, this is achieved through an inefficient means, at least from the network operator's point of view.

6. Proposed Approach

The analysis of most of the existing popular P2P streaming applications, indicate that network efficiency (locality) is not given much attention. Our experimental results showed that Joost's locality is poor and this due to the random connect and disconnect policy, which incurs a negative effect on network locality. On the other hand, Sopcast achieves good levels of network efficiency but only in relatively stable conditions. High levels of churn lead to a considerable degradation in network efficiency. By contrast, Joost is not network efficient due poor locality. Also, a substantial traffic has to originate from servers, to compensate for the best-effort nature of the network and also for poor network locality (the RTT between intercommunicating links is sub-optimal).

Therefore, it is needed to take advantage of the network locality and the load balancing in P2P streaming applications. In this section, a new method for peer-to-peer streaming is proposed and its improved network locality is verified. In addition, QoS and QoE factors were examined and showed additional improvements in different metrics.

6.1 Proposed method

According to our experimental data, we found that a client connects to different peers on the overlay network across the globe. To make clients locally aware, different distance metrics such as RTT, number of hops, and the geographic location may be imposed on connections to maintain the relations between the participant peers. In the proposed method, RTT has been used as a tool to reflect the locality of other peers to any client. A cross-layer approach has then been implemented between the overlay and the underlay network to obtain the RTT values of the participant peers. In our work, though, we have not really focused on new RTT monitoring techniques since this is actually ongoing research topic that is being studied by many authors such as [24] [21]. One way of estimating RTT values is the monitoring method for the intending nodes. Details of various monitoring methods have been published in [24] [25] by one of the authors. Moreover, this is managed by clustering the peers into groups and every cluster is lead by a Super-Node likewise KaZaA. Thus all the queries will go through these Super-nodes instead of tracking all the peers across the network. This strategy helps in avoiding of extra signaling overhead. The aim of this method is to improve the network locality (efficiency), which will be demonstrated in next sections.

6.1.1 Network locality

As mentioned earlier, network efficiency (locality) is the ability to keep the traffic as local as possible, which can be achieved by considering the peers which are nearby with varying the sources among the participants. Therefore, in the proposed method, a decision is made among the participant peers based on the offered RTT values by the monitoring system. Peers are prioritized based on the lower RTT values, and the connections are setup based on the RTT values. Consequently, this will not only maintain the network locality among the inter-communicating nodes but it will also improve the QoS and, hence, the user's quality of experience (QoE).

However, offering network locality only without varying the sources among peers, will be drastically affecting load balancing or in other words, the load distribution between the network and computing sources. Therefore, different techniques are embedded to the proposed method. The main aim of these techniques is to distribute the load among the participants and at the same time having the network locality not affected. This can be shown in the next section.

6.1.2 Computing and network resources load

In order to maintain the load balancing among the contributor peers, different handover techniques have been embedded into the proposed approach. Two conditions trigger handover among interconnected peers:

Switching over: Since the network may experience various constraints such as congestion, bottleneck and link failures, the RTT values will be severely affected and may not be reliable. Additionally, these stochastic conditions will drastically affect the network locality and degrade quality of service (QoS) parameters such as throughput, packet loss, and end-to-end delay. There is also another important requirement arising directly from the adoption of P2P: peers are not reliable entities and cannot be assumed to be always connected. Nodes may leave and join at unpredictable times. So, we must adopt a mechanism which allows receiving peers (in client mode) to maintain a smooth video, although the streaming peers (in server mode) are not constantly available.

One solution to this problem is that any intending client should regularly update the peers' list and re-order them based on the lower RTT values. In our implementation, we keep a ranking list based on RTT distances. Each peer streams to and from 3 other ones. Hence, when this list changes, we apply a switch-over to the new top 3 peers (those with minima RTT to the peer under consideration). This approach has been chosen according to previous findings published in [4] where we found that the average of the active peers that usually a node is downloading from is 3 to 4. Therefore, in this model, the maximum number of sender nodes has been set to be three.

Enforced handover: Another favorable property in the proposed method is the computational efficiency. This can be achieved when the load is periodically distributed among the peers. However, under normal network conditions, peers with lower RTT are selected, but if the condition changes, switch over is applied to the new peers with lower RTT values.

However, some peers may not experience any constraints such as congestion, bottleneck, and link failures. The RTT values will not be affected and may not be changed, so those peers may become the best in every periodical check. Therefore, selecting them regularly would impair the load balancing between the computing and network resources among the peers and the network locality, so an enforced handover is applied.

Furthermore, to avoid pure randomness on the enforced handover process, network locality is applied into clusters of peers, named super-peers, similar to the one adopted in KAZA [27]. Thus, peers are grouped and they are managed by a special peer, or super node. Our experiments have confirmed that the peers on the same cluster share nearly the same RTT values.

7. Performance Evaluation

7.1 Simulation setup

The proposed method has been tested using the ns-2 simulator [27]. The network topology considered for simulations is shown in Figure 8. Moreover, different parameters have been set on for the target topology. First of all, links bandwidth is distributed to all the links evenly as 2Mbps for every link with the same delay; so, all the participants' peers have the same properties. IP as the network protocol and UDP as the transport protocol have been chosen. For simulation of video traffic, the "Paris" sequence of CIF resolution with 4:2:0 formats, was H.264/AVC coded and its packets (chunks) were sent from different peers to the receiver to be assembled on-the-fly by the decoder. These techniques mimic the mesh-based approach on P2PTV streaming. Additionally, to simulate the proposed mechanisms in a more realistic environment, CBR background traffic with UDP transport protocol of packet size of 512-bytes for an average of 1.5Mbps was injected on to the network.

7.2 Simulation scenarios

To verify the proposed method, two scenarios have been implemented, run, evaluated, and compared to each other. The two scenarios have been considered in this paper. The first one considers the proposed method; the second one mimics a typical P2PTV system where peers connect and switchover to and from different peers randomly (*in results section called Randomized approach*), striving for computational load-balancing, as described in [1] [23]. Both scenarios are applied to the same network and loading conditions.

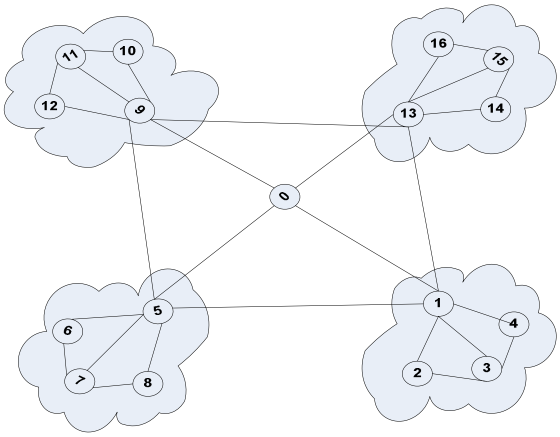


Figure 8 - Simulation Topology

7.3 Evaluation metrics

In order to assess the proposed scheme, different network parameters should be identified, tested and evaluated. The following parameters have been considered:

- Network Locality
- Quality of Service (QoS)
- Quality of Experience (QoE)

8. Results

8.1 Network locality

Network efficiency is the first demand of the proposed method where the inter-connections among the peers should not be done randomly. Thus, it can be seen from figure 9 how the approach is supporting the receiving node to keep the traffic as close as possible, by considering the local peers (*the lower RTT*) at the initiating phase of the connection and in each handover point. So, looking at figure 9, it is clear that the receiving node started with a set of peers which were nearby on the overlay network and achieved almost 0.7 of the network efficiency followed by smooth diminish in few percentages.

This small reduction can be interpreted due to the contribution variation among the participant peers where each peer experiences its own network conditions such as congestion, bottleneck, and sometime link failures. However, the proposed method shows that based on the prioritization among the participants' peers with the lower RTT values and the switching over, network efficiency has increased further (at time 15, 25, and 30) as shown in figure 9.

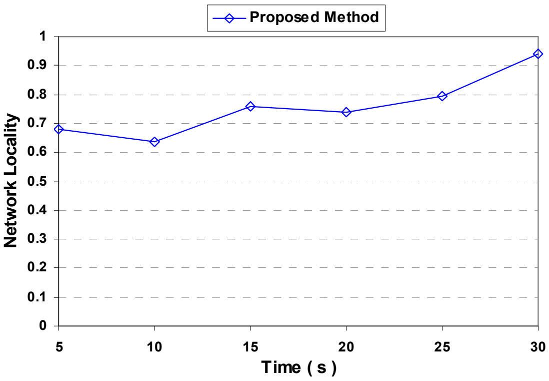


Figure 9 - Network Locality

8.2 Quality of Service

The most important factors that affect the QoS have been considered to determine whether the proposed approach is also actually affecting the quality at the application (or user) level.

In P2P networking, Quality of Service is linked to different metrics about the network. These metrics are intrinsic to each other. For instance, when peers keep downloading from a specific node which offers high bandwidth, but without presenting any sense of balance, the quality of service will be degraded drastically. This also will have other side effects on the computational efficiency (*load distribution*).

For this reason, to quantify and test the proposed method, different effective QoS parameters have been presented and used here as follow:

- Throughput: is the average rate of successful delivery of the packets over the network. The throughput can be measured in bit/s or packet/s.
- Packet loss ratio: This is the ratio between dropped and transmitted data packets. This gives an account of efficiency and the ability of the network to discover routes.
- Average end-to-end delay: The average time span between transmission and arrival of data packets. This includes all possible delays introduced by intermediate nodes for processing and querying of data. End-to-end delay has a detrimental effect on real-time IPTV. This can be countered only to some extent by increasing buffering.
- Delay variation or jitter is another parameter, particularly important for video streaming, as the packets at the receiver need to be decoded at a constant rate. It is desired, if the delay variation to be as minimal as possible.

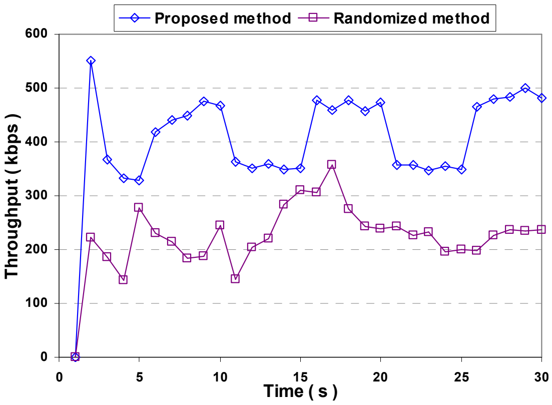


Figure10 - Throughput

Figure 10 shows the effect of the proposed method on the throughput of the two run scenarios. It can be noticed that our local-aware approach in managing the overlay leads to considerable improvements. Our approach reduces the average RTT among inter-communicating nodes and, turns, reduces the overall link utilization. In fact, the average throughput achieved with our approach is 404 kbps, which is almost twice the average throughput obtained by the randomized approach used as a benchmark (210 kbps). If we consider that, according to the literature, P2PTV applications incur a traffic comprised between 300 and 500 kbps [1] [23], we can conclude that our approach brings considerable advantages.

Another parameter which is considered most effective on the QoS is the packet loss rate. This metric is influenced by the congestions on the network, where due to limited buffer capacity overflow packets or packets delayed more than the human perception limit, can be discarded. Figure 11 gives an indication of the advantage of the proposed method since it is trying to reduce the packet loss by switching over among the peers in case of congestion. On the other hand, the randomized approach is showing packet loss and which is due to the randomness in selecting and switching over among the peers.

Packet loss can be considered the most crucial factor in video decoding and, consequently, a determining factor for video Quality of Experience (QoE). It is therefore essential to verify that the P2P overlay is optimized with this regard. Table 1 summarizes findings published in [28], correlating packet loss ratio with video streaming Quality-of-Experience (QoE), for the case of single-layer coding. It is noticeable that any value above around 14% leads to poor quality. According to this table, the proposed method shows a smooth video to the end-users.

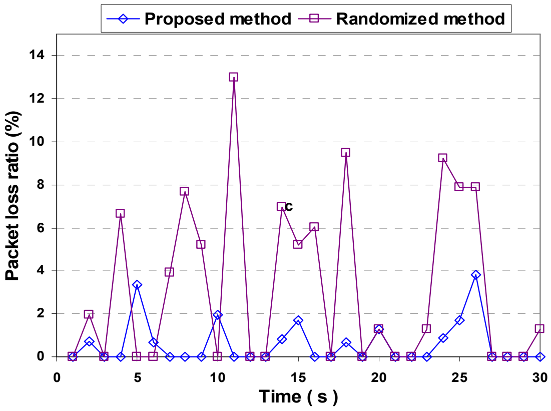


Figure11 - Packets loss ratio

Packet loss ratio [%]	QoE acceptability [%]	Video quality playback
0	84	Smooth
14	61	Brief interruptions
18	41	Frequent interruptions
25	31	Intolerable interruptions
31	0	Stream breaks

Table 1 Quality of experience acceptability thresholds

Another important performance metric in video streaming is end-to-end latency (or delay), which is critical in the process of meeting strict playback deadlines. In turn, this has a direct impact on quality of service and quality of experience.

Figure 12 shows the average end-to-end delays of both scenarios. Noticeably, our approach gives a lower average latency (order of 0.02 seconds); also, we can see that the delay is consistent, due to the fact that connections are not chosen randomly. On the other hand, delay in the randomized approach is fluctuating. This variation in delay may in turn increase packet loss with detrimental effect on QoE.

Figure 13 illustrates this attribute in terms of jitter. A nearly constant jitter derives from the fact that peer handover is governed by a prioritization process (based on RTT values), which ensures that nearby nodes are chosen first. The small variation in jitter in our approach is due to the fact that we still need to force handover even in the case of optimal connections, to maintain a good computational load balance.

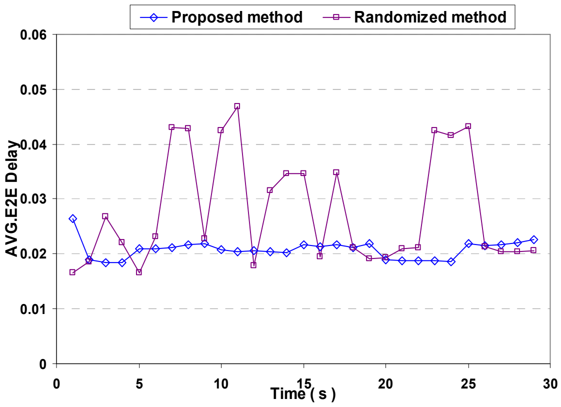


Figure12 - End-to-End Delay

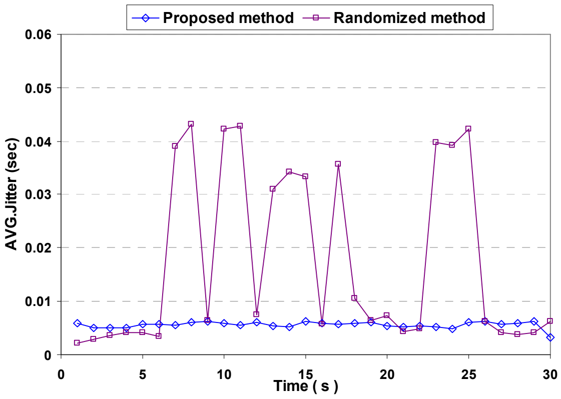


Figure13 - Jitter

8.3 Quality of Experience

Finally, the objective and subjective quality of decoded video under the proposed and randomized methods are compared. The received packets are decoded using error concealment of H.264/AVC encoder JM15. The resulting PSNR is shown in fig. 14 comparing both the algorithms. It is clear that the PSNR improvement is more than 2 dB in most cases, showing the efficiency of our proposed algorithm. The key reason behind the improved PSNR is the smaller degree of packet loss of the proposed method, as shown in fig 11.

Figures 15 and 16 compare the subjective quality of the proposed method against the randomized method respectively, quality of picture on the table, where on the randomized method, the cup, pen, and papers are missing, but in the proposed method, picture quality is considerably better.

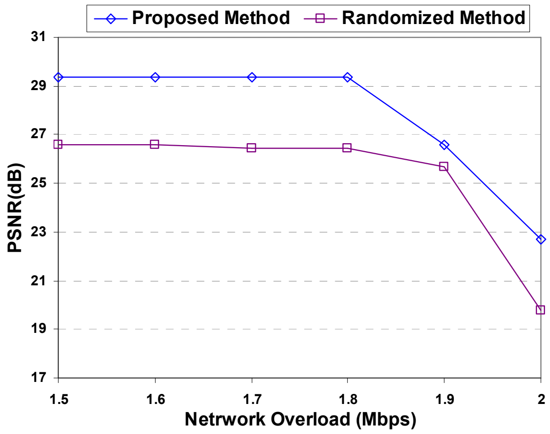


Figure14 – PSNR

6. Concluding Remarks

In this paper, we first evaluate the state of the art in P2P streaming by carrying out a methodical study base on two popular frameworks. Once we identify key shortcomings, we move into an attempt to find simple yet practical and effective ways to address them.

We have looked at two different approaches (Sopcast and Joost), two different streaming models (real-time and VoD), and two different scenarios (peak and ordinary demand levels). In each of the above cases, the aim of the designer was to build a self-managed network overlay that could handle large-scale demand, regardless of fluctuating network conditions and user location.

The P2P paradigm addresses the issue of scalability by limiting the amount of server intervention. As of today, however, P2P application designers do not seem to have placed sufficient emphasis on the need to come up with network-friendly solutions. In this article we have introduced a formula to measure network efficiency in a way that captures network locality (distance among interconnecting peers), link conditions (RTT of those links), and degree of server intervention (ratio between CS and P2P traffic).

A key outcome of our study is that existing approaches do not achieve a good trade-off between computational and network efficiency. Sopcast achieves good levels of network efficiency but only in relatively stable conditions. High levels of churn lead to a considerable degradation in network efficiency. By contrast, Joost is not network efficient due poor locality. Also, a substantial traffic has to originate from servers, to compensate for the best-effort nature of the network and also for poor network locality (the RTT between intercommunicating links is sub-optimal).

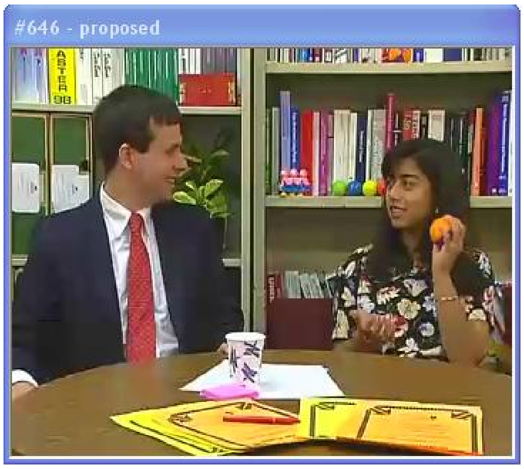


Figure 15 - Video Quality (Proposed)

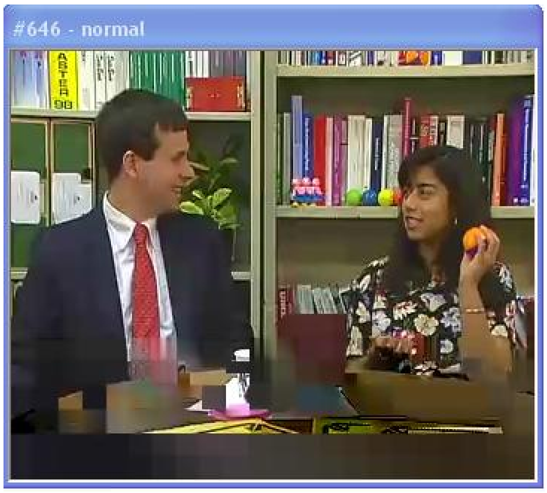


Figure 16 - Video Quality (Randomized)

Existing P2P streaming applications (such as Sopcast and Joost) are appealing from the point of view of the application (or P2P framework) provider as well as the content provider due to the reduced digital distribution costs. However, from the network provider's perspective, such applications do not represent an ideal solution. Problems include:

- Network inefficiency: as demonstrated by our study, network resources are used sub-optimally.
- Traffic aggregation: it is very hard, if not impossible to perform traffic aggregation since traffic sources and destination are not chosen deterministically.
- Manageability: it is very hard to forecast and monitor traffic since user behaviour is unpredictable and traffic sources and destination are not deterministic. Network dimensioning, planning, and control are also difficult.
- Economic models: it is hard to think of a way for the network operator to profit from P2P streaming when most charging models are based on flat rates.

Our proposal is to look for algorithms that employ a certain degree of cross-layer optimization but that retain the simplicity of existing approaches to the best possible extent. In the second part of this article we give an example of a possible way to improve network efficiency and, ultimately, quality of experience.

Our study suggests that the prospects of P2P streaming seem very good if the scalability of this approach is assessed from the application (or even the user) view point. However, if we look from the angle of the network operator the scenario is rather different. Tackling the complex issue of P2P streaming in isolation from network and network management and dimensioning problems is not promising. Simple yet effective approaches based on cross-layer techniques are significantly more powerful.

7. References

[1] M. Alhaisoni, A. Liotta, M. Ghanbari, “An assessment of Self-Managed P2P Streaming”, In Proc. Of ICAS2009 (the 6th International Conference on Autonomic and Autonomous Systems and published by IEEE Computer Society, pp.34-39, 21-24 April 2009 Valencia, Spain.

[2] www.Sopcast.com. Last Access was on 15th of October 2008

[3] www.Joost.com. Last Access was on 27th of October 2008

[4] www.zattoo.com. Last Access was on 20th of October 2008

[5] M. Alhaisoni,, A. Liotta. (2009). Characterization of signaling and traffic in Joost. Peer-to-Peer Networking and Applications, 2(1), 75-83.

[6]

[7] Jun Lei, ,Lei Shi and Xiaoming Fu, (2007)An Experimental Analysis of Joost Peer to Peer VOD Services, Technical Report, Computer Networks Group, University of Gottingen ,Germany.

[8] T. Silverston and O. Fourmaux, “P2P IPTV Measurement: A Comparaison Study”, Universit´e Pierre et Marie Curie, Paris, France.

[9] Ali, S., Mathur, A., and Zhang, H. 2006. "Measurement of commercial peer-to-peer live video streaming". In Proceedings First International Workshop on Recent Advances in Peer-to-Peer Streaming (WRAIPS), August 10, 2006, Waterloo, Ontario, Canada,(pp. 1-12).

[10] Hei,X., Liang,C. ,Liu,Y. and Ross,K.(2007) ‘A measurement stud y of a large scale P2P IPTV system’, IEEE Transactions on Multimedia, Volume: 9, Issue: 8, page(s): 1672-1687

[11] Yong Liu, Yang Guo, Chao Liang, "A survey on peer-to-peer video streaming systems". The Journal of P2P Networking and Applications, Springer, Volume 1, Number 1 / March, 2008, pp. 18-28

[12] Tobias Hoßfeld et al , "A Qualitative Measurement Survey of Popular Internet-based IPTV Systems" 4-6 June 2008, page(s): 156-161

[13] Susu Xie, Bo Li Keung, G.Y. Xinyan Zhang .(2007) ‘Coolstreaming: design, theory ,and practice’ ,IEEE Transactions on Multimedia,Vol.9,No.8,December, page(s): 1661-1671

[14] Y. Liu, X. Liu, L. Xiao, L. M. Ni, and X. Zhang, "BLocation-aware topology matching in P2P systems", in Proc. INFOCOM, Date: 7-11 March 2004 Volume: 4, On page(s): 2220- 2230 vol.4

[15] Z. Xu, C. Tang, and Z. Zhang, “Building topology-aware overlays using globalsoft-state”, in Proc. ICDCS, 19-22 May 2003 On page(s): 500- 508

[16] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, “Scalable application layer multicast”, Dept. of Comput. Sci., Univ. of Maryland, 2002, Tech. Rep. UMIACS.

[17] B. Zhao, A. Joseph, and J. Kubiatowicz, “Locality aware mechanisms for large-scale networks”, in Proc. Workshop Future Directions Distrib. Comput. (FuDiCo), 2002.

[18] Mushtaq, M., Ahmed, T. “P2P Object-based adaptivE Multimedia Streaming”, In J Netw Syst Manage (2007), Volume 15 , Issue 3, Pages: 289 - 310

[19] T. P. Nguyen and A. Zakhor. “Distributed video streaming over Internet”, In Proc. SPIE, Multimedia Computing and Networking., volume 4673, pages 186–195, December 2001.

[20] T. Nguyen and A. Zakhor. “Distributed video streaming with forward error correction”, In Proc. Packet Video Workshop, Pittsburgh, USA, April 2002.

[21] M.Hefeeda, A.Habib, B.Botev, D.Xu, and B.Bhargava (2003), "PROMISE:peer-to-peer media streaming using CollectCast".

In Proc.of ACM Multimedia 2003,pages 45–54, Berkeley ,CA ,USA, November.

[22] Mushtaq, M., Ahmed, T. “Adaptive Packet Video Streaming over P2P Networking Using Active measurements”, In ISCC 2006, Proceeding of the 11th IEEE Symposium on Computers and Communications, pp. 423-428, IEEE Computer Society, Los Alamitos.

[23] Hefeeda ,M.M and Bhargava ,B.K.(2003) ‘On-demand media streaming over the internet’, In Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems, Page: 279

[24] M. Alhaisoni, A. Liotta, M. Ghanbari, “Resource awareness and trade off optimization in P2P Video Streaming", accepted in International Journal of Advance Media Communication, special issue on High-Quality Multimedia Streamingin P2P Environments.

[25] Ragusa C, Liotta A, Pavlou G. “An adaptive clustering approach for the management of dynamic systems”, IEEE Journal on Selected Areas in Communications, 2005; 23(12) 2223–2235

[26] Liotta A, Pavlou G, Knight G. “Exploiting agent mobility for large-scale network monitoring”, IEEE 2002; 16(3), 7–15

[27] KaZaA: http://www.kazaa.com. Last Access was on 27th of Sep. 2009

[28] http://isi.edu/nsnam/ns/ Last Access was on 27th of Sep. 2009

[29] Agboma, F., Smy, M. and Liotta, A. (2008) QoE analysis of a peer to-peer television system. In Proceedings of IADISInt. Conf. on Telecommunications, Networks and Systems, Pages 111-116

Database Connection Monitoring for Component-based Persistence Systems

Uwe Hohenstein and Michael C. Jaeger
Siemens AG, Corporate Research and Technologies
Otto-Hahn-Ring 6, D-81730 Munich, Germany
Email: {uwe.hohenstein|michael.c.jaeger}@siemens.com

***Abstract*—Server applications often use a relational database management system (RDBMS) for storing and managing their data. Storage and management is not limited to the RDBMS itself, but involves also other software forming a persistence system. Such system has become complex. Besides the RDBMS it includes drivers, connection pools, query languages, the mapping between application logic and a database data model and it involves the optimisation of resources. One important resource is the connection from applications to the database system, because the acquisition of a connection is a very expensive operation.**

This work introduces monitoring facilities for the use or misuse of connections in component-based applications. In particular, it explains how the monitoring can take place in order to configure connection pooling for a set of different components. The implemented solution uses conventional programming methods as well as an aspect-oriented approach. The described facilities are integrated into the development of an enterprise-scale application implementing a communications middleware.

***Keywords*-persistence systems; O/R mapping; connection pooling; performance;**

I. Introduction

When it comes to persisting application data, relational database management systems (RDBMSs) are still the most used technology. If the application is written in an object-oriented programming language and a RDBMS is chosen, a mapping between the object data and the relational data model takes place. For this purpose, object-relational (O/R) persistence frameworks exist that implement the orthogonal persistence of data objects [1]. Examples are Hibernate [11], or OpenJPA [22] for the Java platform. By using such a technology, the data model can stay purely object-oriented. Then, programming can be done on an abstract object-oriented level, i.e., operations for storing and retrieving Java objects are provided. The O/R framework translates those object-oriented operations into SQL statements.

In principle, these O/R frameworks ease the work with RDBMSs. However, once the application becomes more complex, once it involves different components and when it processes huge amounts of data, the allocation of resources becomes an important issue for providing optimal performance. An intuitive use of persistence frameworks is not sufficient anymore; the developer must know implications of the used technology and must understand the basic principles such as caching and lazy/eager fetching strategies in order to preserve a high performance level.

Functionality and operations that involve persistence involve also the use of heavy-weight objects or can require the handling of large amounts of data. For example, careless setting of eager fetching can multiply the amount of datasets retrieved. With O/R frameworks and database communication, several heavy-weight objects are used, such as an entity that manages the mapping during run time, or the entity that handles the connection to the RDMS. As a consequence, developers must be aware of settings such as fetching strategies as well as of the proper use of API with regard to the creation of heavy-weight objects.

This work will explain the technical mechanisms managing the connections from the application to the RDMS in the context of a large-scale application middleware developed by Siemens Enterprise Communications (SEN). This middleware implements a service-oriented, server-based platform for common services in the communications domain.

The next Section 2 will explain the particular problems with database connections for component-based systems. The architecture of the SEN middleware is explained in more detail in the next Section 3. Section 4 describes the persistence subsystem of this middleware. Then, the monitoring of connections for this middleware will be explained in Section 5. The presented approach makes use of the recent technology of aspect-orientation (AO) and is applicable to common O/R frameworks such as Hibernate or OpenJPA. The subsequent Section 6 discusses the results of the connection monitoring. The paper will end with our conclusions and future work in this area in Section 7.

II. Common Problems with Database Connections

One important resource that must be handled carefully in large-scale applications is a database connection. A connection object is generally considered a heavy-weight data structure. A database system requires a lot of resources for setting up the communication and to keep space for query results etc. Hence, acquiring and releasing database connections are expensive operations.

A. Amount of Database Connections

In general, the number of connections is often limited by a configuration parameter. Moreover, the underlying operating system, or the RDMS itself has its own limits with regard to a maximum amount of open connections. This means that if one component missed to release a connection, not only resources are wasted, but also other components could be blocked when the overall number of connections is exhausted. O/R frameworks abstract from database connections by offering the concept of a session. A session includes a connection to the database, but does not solve those problems.

Moreover, large applications, having a lot of components and serving a lot of users, require a large number of connections. But in most cases, it is not reasonable to keep connections for a long time. Especially service-oriented architectures (SOAs) are characterised by short-living operations where it does not make sense to hold connections for a long time. Thus, such system has typically a high rate of connection acquisitions.

And also the best practices for O/R frameworks such as Hibernate and OpenJPA suggest using one connection for each transaction and to release it afterwards. This is inconsistent with general recommendations for using JDBC connections, since requesting and releasing a connection is time-consuming. But this is necessary, because each session has an associated object cache that becomes out-of-date at transaction end.

B. Connection Pooling

The efficient handling of short-living connections is supported by connection pools. A connection pool is an entity that allows multiple clients to share a set of connection objects each one providing access to a database. This becomes important, if a large application will serve many clients that each require connection objects. Most O/R frameworks such as Hibernate can be configured to work with a connection pool such as c3p0 [4] or DBCP [7]. Sharing connections in a pool leads to less initialisation attempts of this data structure and thus significantly reduces the time when accessing the RDBMS.

However, a connection pool does not solve all the mentioned problems. It basically reduces the number of required physical connections by means of sharing. Whenever a logical connection is closed, the underlying physical connection is not closed but put back into the pool for further usage. Anyway, closing a logical connection can still be forgotten. Moreover, the parameterization of this pool is not trivial. If the pool contains too few connections, components will have to wait until connections will be released by others. If the pool maintains too many connections, the pool itself, the RDBMS and the operating system will consume unnecessary resources for keeping connections open that are not effectively used.

In order to cope with this case, a pool can be configured to release pooled items after a certain period of inactivity. However, also this feature requires special attention, because if the connection pool might shrink too fast so that new connections must be acquired again; the advantage of a pooled connection is lost. Finally, it is not easy to understand the entire semantics of configuration parameters. Details about these parameters are not scope of this work, but interested readers are advised to compare the behaviour resulting from setting particular DBCP parameters with similarly appearing parameters from c3p0.

C. Issues with Component-Orientation

In a component-based system and also in SOA-based systems, the persistence system is usually provided as an individual instance for each unit of deployment, in most cases for each component or service. This is required, because the O/R mapping information must be given at the persistence systems initialisation time. Otherwise the O/R framework would not know which mapping to perform between objects and relational tables.

As a consequence, also an individual connection pool is maintained for each unit of deployment. Therefore, in a dynamic runtime environment, one can expect different connection pools for each persistence system that covers a domain of O/R mapping definitions. It becomes clear that for a proper use of a connection pool an appropriate parameterization is required in order to ensure optimal performance: Each unit of deployment must provide sufficient connections for the highest throughput, but otherwise just as few connections as possible to allow appropriate settings for other connection pools as well.

Configuring one connection pool is difficult, because it requires an appropriate load model and also sufficient measurements facilities. Besides the number of used connections, it must be also tracked, how long a component or thread is blocked when obtaining a connection. Considering a component-based system, it becomes clear that configuring several pools is even more difficult: How can we obtain an appropriate load model that also resembles the required parallelism?

Moreover, services call each other, which implicitly relate their connection pools somehow implicitly. Even if the connection pool of a service is large enough for its purpose, performance is degraded if the service is calling another service the pool of which is congested. It is also clear that connection information from the RDBMS is not sufficient, because the RDBMS does usually not provide information about the originating component for a connection. Such information can only be obtained by intercepting JDBC drivers or the internals of O/R mapping frameworks.

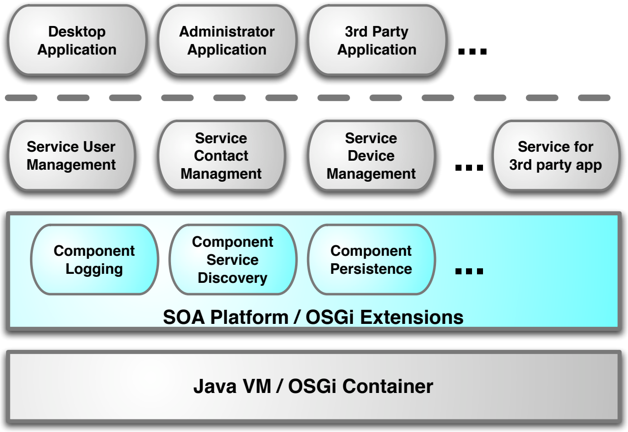

Desktop Application
Administrator Application
3rd Party Application
...
Service User Management
Service Contact Managment
Service Device Management
...
Service for 3rd party app
Component Logging
Component Service Discovery
Component Persistence
...
SOA Platform / OSGi Extensions
Java VM / OSGi Container

Figure 1. OpenSOA Basic Architecture

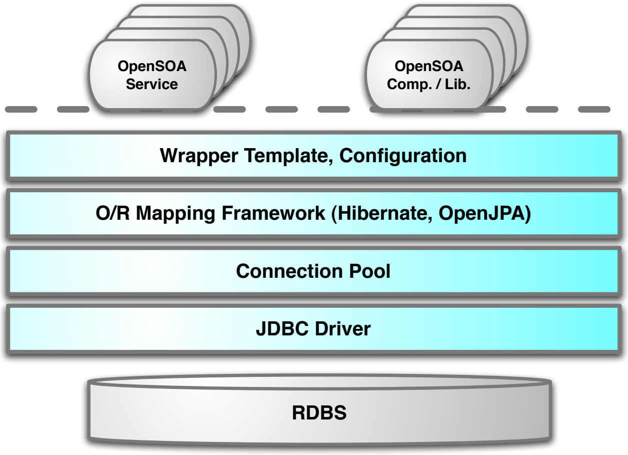

OpenSOA Service
OpenSOA Comp. / Lib.
Wrapper Template, Configuration
O/R Mapping Framework (Hibernate, OpenJPA)
Connection Pool
JDBC Driver
RDBS

Figure 2. Persistence System Architecture

III. The OpenSOA Middleware

The application case where the monitoring was applied is a middleware called OpenSOA [28]. OpenSOA is implemented in Java and provides an open service platform for the deployment and provision of communication services. Examples for such services are the capturing of user presence, the management of calling domains, notifications, an administration functionality for the underlying switch technology, and so forth. One business case is to sell these services for the integration in groupware and other communication applications along with the Siemens private branch exchange (PBX) solutions. The technical basis for OpenSOA is an OSGi container [23]. This specification was chosen over JEE [8] because of its smaller size and focused functional extent.

In order to establish an infrastructure for the provision of services, custom built or common off-the-shelf components (depending on the availability) were added. For example, they implement the discovery and registration of services among containers on different computers for a high-performance messaging infrastructure. Figure 1 outlines the basic architecture: as the foundation, the software runs on Java and an OSGi container. Then, a tier provides components and extensions in order to implement a service-oriented architecture (SOA). On top of that, different services implement common functionality of a communication system. Finally, different application can communicate with this system using service interfaces.

The size of the entire code base (infrastructure and communication services) has similar dimensions as the distribution of the JBoss application server with regard to the defined classes (in the range of 5.000 to 10.000). It involves about 170 sub-projects among which about 20 projects use the persistence system.

IV. OpenSOA Persistence System

The previous overview OpenSOA did not explain the persistence system of OpenSOA which is subject of this section. Figure 2 shows the persistence architecture as an outline. The persistence system is divided into different tiers. Each of the tiers accesses the functionality provided by the tier below. An OpenSOA service using persistence services generally accesses the wrapping template.

A direct access of the O/R mapping framework or JDBC is not desired, however, cannot be prevented. The O/R mapping framework obtains connection from a connection pool. And the pool accesses the database via the JDBC driver. Again, the direct access to the database cannot be prevented by using the JDBC driver. But in practise it is a set rule that such calls are forbidden.

In the beginning, the Hibernate framework was used as implementation of the O/R mapping. Then, because of a patent infringement claim against Red Hat (the vendor of Hibernate), the persistence system was migrated to OpenJPA distributed by the Apache Software Foundation.

The entire persistence system allows OpenSOA for running on several RBDSs. Various business reasons require the support of solidDB from IBM: solidDB provides high performance paired with high reliability because of its hot-stand-by feature. In addition, MySQL and PostgreSQL are also supported. Moreover, the persistence framework makes programming easier by offering an object-relational mapping in order to store and retrieve the data of Java objects.

A. Working with O/R Mapping

An O/R mapping framework like Hibernate or OpenJPA requires the information about which Java classes and which fields in them are persistent, how classes map to tables, and how fields map to table columns and how pointers are mapped to foreign keys. OpenJPA supports the definition of mappings by using an XML-document or by annotation in the Java code (with Java 5). In JPA, a self-contained set of mapping definitions forms a so called persistence unit.

The APIs of Hibernate and OpenJPA are very similar. Both use a factory object that maintains connection objects to the storage. The connection object is called Session in Hibernate and EntityManager in the JPA specification. Accordingly the factory objects are called SessionFactory and EntityManagerFactory respectively. An EntityManager-Factory exists for each persistence unit. Consequently, a connection pool is by default maintained for each persistence unit.

For programming actual operations, the developers needs to obtain a Session or an EntityManager object. From a conceptual point of view, a connection to the database is opened then (Actually, OpenJPA, for example, can be configured to actually defer the opening of a connection in order to tune for a shorter period of connection obtainment). The O/R mapping framework obtains database connections from a connection pool implementation, while the pool obtains concrete database connection from the JDBC driver. In addition, a connection pool such as DBCP or c3p0 is set between the JDBC driver and the persistence framework. The JDBC driver is used to communicate with the RDBMS.

A general recommended programming practice is to begin a transaction as well at this point. When a session is opened, the developer can perform various object operations, such as creation, deletion, modifications that are translated to according SQL statements. If a set of operations is finished, the developer should commit the transaction and close the session. From a conceptual point of view, the connection to the database is closed then. In conjunction with a connection pool, the connection is free for the next connection acquisition then. Listing 1 shows an example use with OpenJPA.

Listing 1. Example Use of OpenJPA

```
EntityManagerFactory emf =
Persistence.createEntityManagerFactory("myPerUnit");
 EntityManager em = emf.createEntityManager();
 EntityTransaction tx = em.getTransaction();
 Customer c = new Customer(4711);
 em.persist(c);
 Query q =
    em.createQuery("SELECT␣c␣FROM␣Customer␣c");
 List result = q.list();
 Customer c2 = mySession.find(Customer.class, 42);
 tx.commit();
em.close();
```

B. Wrapping Template

The OpenSOA middleware extends the persistence framework with a wrapping template implementation, which standardises the use of Hibernate or OpenJPA. This template works in a similar way as the JDBC abstraction in the Spring framework [26]. It offers different functionality upon the O/R mapping framework. It basically abstracts from concepts such as OpenJPAs EntityManagerFactory (in principle a database with a schema and an associated connection pool), EntityManager (a database connection) and persistence units (logical database name). It ensures the appropriate and standardised use of this framework. When a developer uses the persistence system via the wrapping template, the following functions are provided:

1) Standardised parameterization of the persistence framework. This includes the RDBMS URL, the access credentials as well as other settings performed by the framework.
2) Standardised allocation of resources. A general setting for opening and closing connections and transactions is performed.
3) Standardised error handling. In case of recoverable errors, exceptions returned by the persistence framework or by the JDBC driver are caught and a retry is initiated, e.g., in case of concurrency conflicts. Furthermore, exceptions are converted into OpenSOA-specific ones for coverage in the layers above.

Although it is theoretically possible for developers to create, for example for the OpenJPA case, an EntityManager directly, this is generally a not allowed practice. The subsequent discussion will focus on OpenJPA and solidDB although the principles have been applied to the Hibernate framework and the other RDBMSs as well.

V. Connection Monitoring

The general aim of the monitoring is the proper configuration and usage of connections and the connection pool. Main pool parameters are the initial number of connections, the maximum and minimum number possible and the idle time after which a connection is closed after inactivity. It is a complex task to determine appropriate settings for the pool. Moreover, it is important to monitor the correct usage of connections, e.g., to avoid that connections are not released.

The monitoring functionality has been implemented at different levels in order to cover the different ways a client to the persistence system can obtain connections. Generally, the developer of a service should implicitly obtain connections by using the wrapping template. That is the first place to integrate monitoring. However, this corresponds to a logical acquisition of connections since not always real physical connections are requested thanks to pooling. The overall goal is also to get information about any physical open/close connection activity in order to keep track of the number of currently used connections for each functional unit. However, there is major a problem to obtain this information: in principle, the JDBC driver needs to be intercepted, but for example in case of solidDB, its source code is not available. To this end, the recent technology of aspect-oriented programming AOP provides an adequate solution.

A. Aspect-Oriented Programming with AspectJ

Aspect-oriented programming (AOP) provides a solution, which is not immediately obvious. AOP has originally been proposed for developing software to eliminate crosscutting concerns (CCCs) [10]. Those CCCs are functionalities that are typically spread over several classes, thus leading to code tangling and scattering [5], [20] in conventional programming. AOP provides special extensions to Java to separate crosscutting concerns. Recent research has shown usefulness to this respect, e.g., [12], [25], [29], [13], [19].

AOP is ideal for monitoring purposes. So far, some tools are implemented with AOP to monitor the system performance [3], [9]. We applied the AO language AspectJ [18], [16] to monitor the usage of database connections.

Programming with AspectJ is essentially done by Java and newly aspects. An aspect concentrates crosscutting functionality. The main purpose of aspects is to change the dynamic structure of a program by changing the program flow. An aspect can intercept certain points of the program flow, called join points. Examples of join points are method and constructor calls or executions, attribute accesses, and exception handling. Join points are syntactically specified by means of pointcuts. Pointcuts identify join points in the program flow by means of a signature expression. Once join points are captured, advices specify weaving rules involving those joint points, such as taking a certain action before or after the join points.

As AspectJ is a new language, which offers no syntactic constructs such as aspect, pointcut and advice, it requires a compiler of its own. Usually, the AJDT plug-in will be installed in Eclipse. However, a new compiler requires changes in the build process, which is often not desired. Then, using Java-5 annotations is an alternative: Aspect code can be written in pure Java; we could rely on standard Eclipse with an ordinary Java compiler, without AJDT plug-in for AspectJ compilation etc. The Listing 2 is an example that uses Java classes with AspectJ annotations.

Listing 2. Example Use of OpenJPA

```
@Aspect
class MyAspect {
 internal variables and methods;
 @Before(execution(* MyClass*.get*(..)) )
  public void myPC() {
    do something before join point
  } ... }
```

An annotation @Aspect lets a Java class MyAspect become an aspect. If a method is annotated with @Before, @After etc., it becomes an advice that is executed before or after join points, resp. Those annotations specify pointcuts as a String. This aspect possesses a @Before advice that adds logic before executing those methods that are captured by the pointcut myPC. In order to use annotations, the AspectJ runtime JAR is required in the classpath. To make the aspect active, we also have to start the JVM (e.g., in Eclipse or an OSGi container) with an javaagent argument referring to the AspectJ weaver. Annotations are then evaluated and become really active, because a so-called load-time weaving takes place: Aspect weaving occurs whenever a matching class is loaded.

If loadtime weaving cannot be applied, e.g., if AspectJ should be applied to code running in an OSGi container, another pre-compilation approach can be used which requires the iajc compiler. We can take the aspect class and compile it into a JAR file with an ordinary Java compiler. Then, the aspects JAR can be applied to classes or existing JARs, particularly, 3rd party JARs such as Hibernate or JDBC drivers. There is an iajc taskdef in ANT to make both steps easier. That is the approach we are pursuing. This is only a brief overview of AspectJ; examples are given later in the code samples.

B. Monitoring Component

All the connection information is collected by a central persistence statistics component. This component implements an OSGi component and acts as a subpart of the persistence system. This component offers two basic interface parts:

One part provides the parameterisation of the statistics functionality. It covers switching certain functionality on and off. And it defines how detailed the monitoring results should be. The output is written to a log with a definable frequency.

The other part receives the notifications of various connection attempts to obtain and release connections. The component receives such notifications and saves this information in a counter-based local data structure. Since during the life cycle, the entire application uses millions of connections, a comprehensive logging of every attempt would not make sense. Hence, the persistence statistics tracks for only currently open entities and stores an operational maximum number in addition.

Furthermore, the persistence statistics component can place an alert, if a connection remains open without any closing action, or if an attempt to close a non-existent connection happens (where no preceding obtainment has taken place). It is important to determine the origin of connection acquisitions, which could be resembled by the database user or by the persistence unit. In our case, there are only few database users are configured. Extracting the persistence unit is a better choice.

The data for these events is also separated by the functional components. That is, the persistence statistics stores all attempts separated by the user management services, all by the contact list service, etc. Depending on the parameterisation of the persistence statistics component, reports of this data can be written at different levels of granularity at different time intervals.

C. Connection Pool Monitoring: Wrapper Template

As already mentioned, the persistence system introduces a wrapping template that encapsulates database requests at a central place. This template is a first place to add monitoring functionality. It covers the notification of following events: First, the wrapping template notifies the obtainment and the release of an EntityManager of OpenJPA (which is a Session in Hibernate). Then, the template notifies the persistence statistics component when such a logical connection is obtained or released. Depending on the parameterisation, the persistence framework can automatically initialise a connection when a Session/EntityManager is obtained, or it can defer this step to later phases. For example, OpenJPA allows for three settings of connection initialisation:

a) a connection is automatically obtained for an obtained EntityManager,
b) a connection is obtained if a transaction is opened and
c) a connection is opened on demand when a connection is actually used by the persistence framework.

This represents the acquisition of logical connections. Due to a connection pool, this does not necessarily acquire a new physical JDBC connection. Similarly, closing an EntityManager is handled. The Listing 3 presents an excerpt of the template implementation.

Listing 3. Control Points in Operation Flow

```
start = System.currentTimeMillis();
session = sessionFactory.openSession();
if  (OpenJPAWrapper.getStatistics()
  .isNotificationsGenerallyEnabledYN()) {
 EntityManagerFactory emf =
   this.sessionFactory.getEMF();
 persistenceUnit = (String)
   emf.getConfiguration()
     .toProperties(false).get("openjpa.Id");
 OpenJPAWrapper.getStatistics()
     .notifySessionOpened(
        persistenceUnit, getActionId());
}
...
EntityManager oem = session.getEM();
trans = session.beginTransaction();
if (OpenJPAWrapper.getStatistics()
  .isNotificationsGenerallyEnabledYN()) {
    connectionId = jdbcConn.hashCode();
  OpenJPAWrapper.getStatistics()
    .notifyTemplateConnectionObtain(
      persistenceUnit, getActionId(),
        connectionId, elapsed);
}
stop = System.currentTimeMillis();
elapsed = stop   start;
if (elapsed > THRESHOLD) {
  printTimingWarning("connection",
       action, sql, elapsed);
}
start = System.currentTimeMillis();

if (OpenJPAWrapper.getStatistics()
  .isNotificationsGenerallyEnabledYN()) {
    OpenJPAWrapper.getStatistics()
      .notifyTemplateConnectionRelease(
persistenceUnit, getActionId(), connectionId);
OpenJPAWrapper.getStatistics()
      .notifySessionClosed(
           persistenceUnit, getActionId());
}
session.close();
```

D. Connection Pool Monitoring: JDBC Driver

The overall goal is also to get information about any physical open/close connection activity in order to keep track of the number of currently used connections for each functional unit. Since the JDBC driver needs to be intercepted the source code of which is generally not available, we apply the monitoring aspect as listed in Listing 4.

Listing 4. Aspect for JDBC Driver: Origin

```
@Aspect
public class ConnectionMonitorAspect {
 @AfterReturning(
   pointcut = "execution(Connection
solid.jdbc.SolidDriver.connect(..))",
   returning = "ret")
  public void monitorOpenJDBC
  (final Connection ret) {
   if (monitoringIsPossible) {
    String unit = determineFunctionalUnit();
    SolidConnection conn =
                   (SolidConnection) ret;
    theStatistics.notifyJDBCConnectionObtain
                      (unit, conn.hashCode());
  } } ...}
```

At a first glance, ConnectionMonitorAspect is a Java class that possesses a method named monitorOpenJDBC. However, an annotation @Aspect lets the Java class become an aspect. Since the method is annotated with @AfterReturning, it becomes an advice that is executed after returning from a method; the returning clause binds a variable ret to the return value. The method possesses a corresponding parameter Connection ret that yields access to the return value.

The @AfterReturning annotation also specifies the pointcuts as a String. Here, any execution (execution) of a method SolidDriver.connect with any parameters (..) returning a Connection is intercepted and the logic of the monitorOpenJDBC method is executed after being returned. In other words, this aspect monitors whenever a connection is opened. It determines the functional unit and uses the hashCode to identify the connection and pass both to the persistence statistics by using theStatistics.notifyJDBCConnectionObtain.

There are other forms of advices such as @Before (executing before) or @Around (putting logic around or replacing the original logic) which are handled similarly. Closing a connection via JDBC is monitored by the aspect shown in Listing 5.

Listing 5. Aspect for JDBC Driver: Counting

```
@Before("execution(*
solid.jdbc.SolidConnection.close(..))")
public void monitorCloseJdbc (final JoinPoint jp) {
  if (monitoringIsPossible) {
    String unit = determineFunctionalUnit();
    SolidConnection conn =
              (SolidConnection)jp.getThis();
    theStatistics.notifyJDBCConnectionRelease
                   (unit, conn.hashCode());
} }
```

The parameter JoinPoint jp gives access to context information about the join point, especially the object on which the method is invoked (jp.getThis()). The gathered information is sent to the central theStatistics object that keeps this information to determine the number of currently open connections. Moreover, it keeps track of the maximal value. This information is managed for each functional unit. The important point is that the solidDB JDBC driver is intercepted although its source code is not available. That is, the aspect code is applied to a 3rd party JAR file.

E. Detecting Misusage

Developers should use the template to access the database, however, there is no way to force them. Sometimes developers use JDBC to connect to the RDBMS directly, thus bypassing the entire infrastructure, e.g., for administration or setup purposes. When it comes to the monitoring of connections, all the attempts from different levels of the persistence system architecture must be monitored: Directly using JDBC connections has the inherent danger of programmers not releasing the connection, which would lead to an increase in connection usage.

Additional aspect advices allow for monitoring certain kinds of misusage when using the persistence framework directly. For example, it is monitored whenever an EntityManagerFactory is created by bypassing our wrapper template (creation is usually not done explicitly, but implicitly in the template). This means that an additional connection pool would be created for the same functional unit. The corresponding pointcut is:

```
@Before("execution(
   * *..*.createEntityManagerFactory
  (String, java.util.Map)) && args(str,map)")
```

Another pointcut detects any direct usage of JDBC connection handling beside the connection pool:

```
@Pointcut("execution(*
    org.apache.commons.dbcp.*.*(..))")
public void withinDbcp() {}
@AfterReturning(pointcut = "execution(*
    solid.jdbc.SolidDriver.connect(..))
        && !cflow(withinDbcp)", returning = "ret")
```

In both cases, an advice will issue a warning.

F. Connection Aquisition Times

Besides the amount of actual open connections and also besides the information of the origin, a problem remains: From the counting of connections, it cannot be seen, if the pool actually performs efficiently when providing connection objects. It is obvious that connection counting can happen only at specified intervals. However, an overload situation can easily slip through the measurement points.

Thus, another measurement facility has been added to the wrapping template, a time measurement, how long actually a connection obtainment takes place. If the obtainment takes place obviously the demand is higher than the number of connections that the connection pool provides.

In addition, we can guess at medium-ranged connection obtain times that the pool has freshly created the connection. Please note that it depends on the computer system used how many milliseconds the initialization of a connection object actually takes. This can have two reasons: Either the settings of the pool reduce the number of kept open connections too quickly. Or the integrity conditions for a connection are missed too often, which lets the pool replace an existing connection with a new one. Both cases also reduce the performance of the overall system. In the first case, the settings of the pool need revision. In the second case, the connection integrity conditions must be checked or the handling of connections must be evaluated for inappropriate operations on the connection object.

Although we issue a warning whenever the time to obtain a connection exceeded a certain threshold, we still do not know the reason why: Is it because the pool is exhausted and the DBS has to create a new one? Or has the connection pool some contention problems solving parallel requests? To get more diagnostics, we added an additional aspect that allows distinguishing whether a connection is newly created or taken from the pool, thereby passing a threshold. In addition, the current connection pool properties (recently active and idle connections) are printed.

G. Summary

The proposed environment offers the desired information about connections. It presents an overview of currently requested logical database connections and really used physical connections for any component a certain time intervals. Moreover, it keeps track of maximal values. If the logical number value is higher than the physical one, then the pool seems to be undersized. Similarly, an oversized pool can easily be detected. An example output send top the log files is listed in Listing 6.

Listing 6. Example Output of Monitoring

```
14:12:00,097 DEBUG [PersistenceStatistics]
  @b6d6ab says: sess opened ever/current: (1/1),
   template conn obtain
      ever/max at once/current open: (1/1/1),
   jdbc conn max same time/current: (35/35).
...
14:12:00,097 DEBUG [PersistenceStatistics]
 session referrer history:
--------------------------------------
domain:
 |_ com.siemens...createDomain(),
       ever: 1, curr: 1
   |_ max 1 session(s) in use at the same time.
--------------------------------------
```

```
 connection referrer history:
---------------------------------------
domain:
 |_ com.siemens...createDomain(),
       ever: 1, curr: 1
   |_ max 1 template conn(s) at the same time.
---------------------------------------
 1 current template connections (max at once: 1):
---------------------------------------
  12743356 origin: com.siemens...createDomain(),
---------------------------------------
 35 current jdbc connections (max at once: 35):
---------------------------------------
 |_ domain: 35
---------------------------------------
...
```

In addition to the introduced monitoring in the previous sections, also the RDBMS is queried for the connected users and the number of open connections. Figure 3 shows a comprehensive overview of the monitoring points in the architecture.

VI. Results

The introduced connection monitoring allowed insights on the use of connections for particular services or components. Such kind of information is important for the following reasons:

1) Some functionality is called very seldom, resulting in few database operations, for example a backup service, while other functionality is called in a continuous manner, for example, a user management service in several threads. The distinction by the functional units allows for an assessment of required parallel connections to the RDBMS. Over- and under-sizing can be avoided.
2) The OpenSOA platform and the services will be combined in different deployment scenarios resulting in different total amount of required connections. Such aggregated numbers are important for the proper parameterisation of the RDBMS, or the server running the RDBMS.
3) The monitoring of JDBC connections also verifies the proper use of the persistence framework. When using a persistence framework, avoiding direct connections to the RDBMS is very important. Otherwise, the object initialization, caching and the retry mechanisms can lead to inconsistent data. Using our monitoring, we could detect one component that uses JDBC directly.
4) The monitoring revealed that the number of connections oscillates within seconds with some settings of the DBCP: sometimes, even if components were waiting for a connection from DBCP, newly released connections were physically closed and immediately requested again because of the load. This is an important performance issue that leaded to severe performance degradation. Using our monitoring, it turned out that the maxActive configuration parameter defines the upper limit of connections. If another maxIdle parameter is lower, then DBCP immediately releases any exceeding connection - even if there are still requests!
   Furthermore, idle connections are evicted in a configurable interval until a minIdle threshold is passed. Again some kind of oscillation occurs, since a bulk of connections is released first, and then connections are acquired to satisfy the minIdle parameter. This strange and unexpected behavior of the DBCP could then be fixed.
5) If one component calls additionally internal operations within the same persistence unit and thus the same pool a deadlock situation can occur: At high loads the maximum amount of possible open connection can be reached by originating calls. Then a sub operation required cannot obtain further connections. However, the originating operation cannot proceed, because it waits for the sub operation. Thus, the combination of appropriate load models for testing and detailed monitoring facilities is required to systematically identify such deadlock situations.

The applied connection monitoring has clearly improved the knowledge about the deployment needs of the OpenSOA platform. We ask the reader for understanding that no detailed figures can be given as the OpenSOA platform is commercial software acting in competition to other solutions. Regarding related solutions, it must be noted that the presented implementation of such a monitoring was required for the following reasons:

1) Not every connection pool offers this information at the required grade: For example, the DBCP connection pool that we integrated into OpenJPA offers no logging at all in the current version.
2) The solidDB JDBC driver does not offer such logging facilities as required.
3) Even if some of the information were available with OpenJPA, Hibernate, or if an alternative connection pool like c3p0 would provide the information here and there, the analysis would require a cumbersome search inside the log files. In contrast, the a centralised persistence statistics component offers also the advantage of an aggregated reporting.
4) A RDBMS stores the information at system level. However, such information can be only used to verify the information gathered from the various other points. The RDBMS stores in most cases information about the number of connections obtained by a particular database user. Since different components of an application use the same database user, no detailed information can be found from there.

VII. Conclusions

Monitoring database connections in complex software is manifold and cannot be done by the database system only,

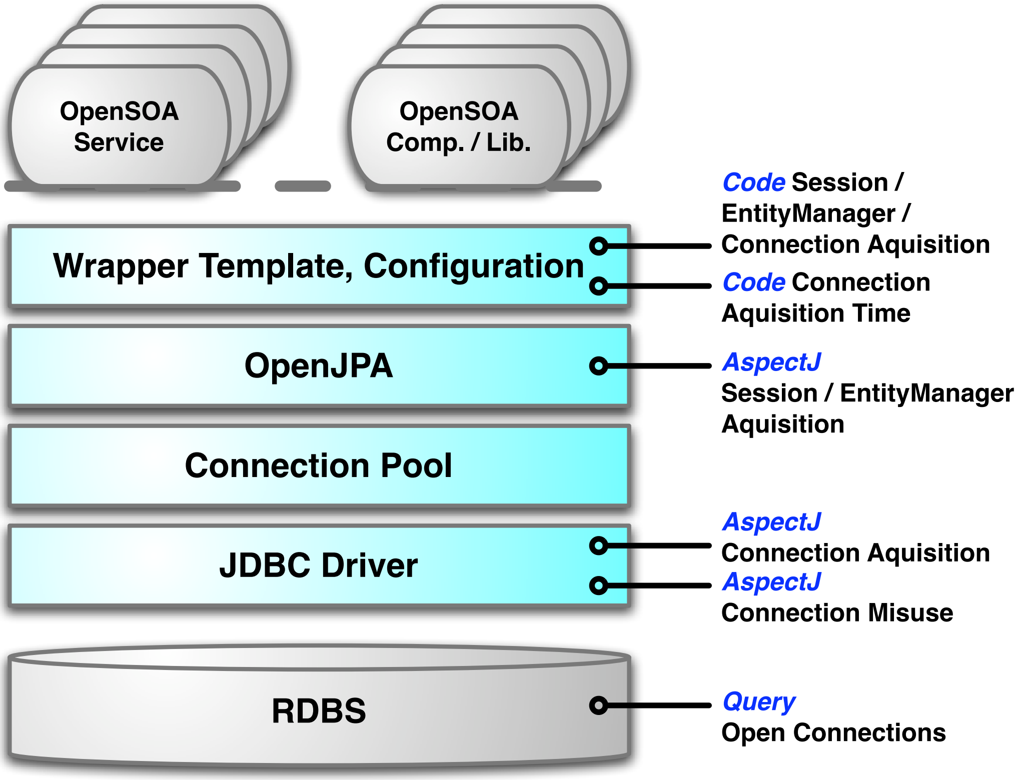

OpenSOA Service
OpenSOA Comp. / Lib.
Wrapper Template, Configuration
OpenJPA
Connection Pool
JDBC Driver
RDBS
Code Session / EntityManager / Connection Aquisition
Code Connection Aquisition Time
AspectJ Session / EntityManager Aquisition
AspectJ Connection Aquisition
AspectJ Connection Misuse
Query Open Connections

Figure 3. Monitoring in the Persistence System

if the application is distributed and involves many different components. Moreover, statistics over the counted connections is mandatory in order to precisely define parameters for optimal resource consumption/performance trade-off of the connection pool. The discussion has motivated that a precise monitoring of connections to the database and their origin is relevant in order to optimise different deployment configurations in terms of the parameters for a connection pool.

As future work, a self-parameterisation of the connection pool parameters can be implemented based on the monitoring functionality. For this purpose, dynamic connection pools exist already. However, a combination with the dynamic deployment of bundles in the OSGi container would improve adaptation of connection pool parameters instead of monitoring the resulting traffic on the connection pool level only. Such a mechanism requires additional research, because the optimal number of connections would result from a statistical model that includes also usage patterns and the number of served users.

References

[1] M.P. Atkinson, R.Morrison: Orthogonally Persistent Object Systems. VLDB Journal 4, 3 (1995) pp 319-401.

[2] M. Aksit (ed.): Proc. of 2nd Int. Conf. on Aspect-Oriented Software Development, Boston 2003

[3] R. Bodkin: AOP@Work: Performance monitoring with AspectJ. http://www.ibm.com/developerworks/java/library/j-aopwork10/index.html

[4] c3p0 Connection Pool, Project site available at: http://c3p0.sourceforge.net/

[5] Y. Coady, G. Kiczales: Back to the Future: A Retrospective Study of Aspect Evolution in Operating System Code. In [AOSD03]

[6] M. Chapman, A. Vasseur, G. Kniesel (eds.): Proc. Of Industry Track 3rd Conf. on Aspect-Oriented Software Development, AOSD 2006, Bonn, ACM Press

[7] Commons DBCP Component. project website available at http://commons.apache.org/dbcp/

[8] The EJB3 Specification (JSR 220): http://java.sun.com /products/ejb/docs.html

[9] K. Govindraj, S. Narayanan et al.: On Using AOP for Application Performance Management. In [CVK06]

[10] R. Filman, D. Friedman: Aspect-Oriented Programming is Quantification and Obliviousness. Worksh. on Advanced Separation of Concerns, OOPSLA 2000

[11] Hibernate Reference Documentation. http://www.hibernate.org /hib_docs/v3/reference/en/html/

[12] J. Hannemann and G. Kiczales: Design Pattern Implementation in Java and AspectJ. In Proc. of the 17th Annual ACM Conf. on Obj.-Orien. Programming, Systems, Languages, and Applications, OOPSLA 2002, Seattle

[13] U. Hohenstein: Using Aspect-Orientation to Manage Database Statistics. In:

[14] JSR-000012 JavaTM Data Objects Specification. http: //jcp.org/aboutJava/communityprocess/first/jsr012/

[15] Java Persistence API. http://java.sun.com/javaee/technologies /persistence.jsp

[16] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, W. Griswold: An Overview of AspectJ. ECOOP 2001, Springer LNCS 2072

[17] G. Kiczales: Adopting AOP. In Proc. 4th Conf. on Aspect-Oriented Software Development; AOSD 2005, Chicago, ACM Press

[18] R. Laddad: AspectJ in Action. Manning Publications Greenwich 2003

[19] N. Lesiecki: Applying AspectJ to J2EE Application Development. IEEE Software, January/February 2006

[20] G. Murphy, A. R. Walker, M. Robillard: Separating Features in Source Code: An Exploratory Study. In Proc. of 23rd Int. Conf. on Software Engineering 2001

[21] H. Masuhara, A. Rashid (eds.): Proc of 5th Int. Conf. on Aspect-Oriented Software Development. Bonn (Germany) 2006

[22] Java Persistence API. http://java.sun.com/javaee/ technologies /persistence.jsp

[23] Open Service Gateway Initiative. Home Page of the OSGi Consortium. http://www.osgi.org.

[24] Rashid, A.: Aspect-Oriented Database Systems. Springer Berlin Heidelberg 2004

[25] A: Rashid, R. Chitchyan: Persistence as an Aspect. In [AOSD03]

[26] Rod Johnson: Introduction to the Spring Framework. October 2007. http://www.theserverside.com/tt/articles/ article.tss?l=IntrotoSpring25

[27] S. Soares, P. Borba: Implementing Modular and Reusable Aspect-Oriented Concurrency Control with AspectJ: In WASP05, UberlŁndia, Brazil

[28] W. Strunk: The Symphonia Product-Line. Java and Obj.-Orient. (JAOO) Conf. 2007, Arhus, Denmark (2007)

[29] C. Zhang, H.-A. Jacobsen: Quantifying Aspects in Middleware Platforms. In [AOSD03].

Run-Time Connector Synthesis for Autonomic Systems of Systems

Radu Calinescu
Computing Laboratory, University of Oxford, UK
Radu.Calinescu@comlab.ox.ac.uk

***Abstract*—A key objective of autonomic computing is to reduce the cost and expertise required for the management of complex IT systems. As a growing number of these systems are implemented as hierarchies or federations of lower-level systems, techniques that support the development of *autonomic systems of systems* are required. This article introduces one such technique, which involves the run-time synthesis of *autonomic system connectors*. These connectors are specified by means of a new type of autonomic computing policy termed a *resource-definition policy*, and enable the dynamic realisation of collections of collaborating autonomic systems, as envisaged by the original vision of autonomic computing. We propose a framework for the formal specification of autonomic computing policies, and use it to define the new policy type and to describe its application to the development of autonomic system of systems. To validate the approach, we present a sample data-centre application that was built using connectors synthesised from resource-definition policies.**

***Keywords*—autonomic computing; systems of systems; autonomic system connectors; model-driven development; automated code generation.**

I. Introduction

The original vision of autonomic computing was one of self-managing systems comprising multiple inter-operating, collaborating *autonomic elements* [2]. Like the components of a biological system, these autonomic elements were envisaged to contain well-differentiated *resources* and deliver *services* to each other, to integrate seamlessly, and to work together towards a global set of system-level objectives [2][3][4].

This view of self-managing systems as collections of collaborating autonomic elements is entirely consistent with the way in which many of the most important computing systems are designed, built and managed. Indeed, key computing systems in areas ranging from transportation and healthcare to aerospace and defence can no longer be implemented as monolithic systems. Instead, the capabilities and functionality that these systems are required to provide can only be achieved through employing collections of collaborative, heterogeneous and autonomously-operating systems [5][6].

Thus, the characteristics of the computing systems that would benefit most from self-managing capabilities, and the vision behind the sustained research efforts in the area of autonomic computing are well aligned. Nevertheless, the same is hardly true about the outcome of these efforts: what the overwhelming majority of the autonomic computing research to date has produced is a collection of methods, tools and components for the development of *monolithic* autonomic systems; and exemplars of such systems. The few notable exceptions (overviewed in Section II) are either high-level architectures [7] that refine the "collection-of-autonomic-elements" architecture from the seminal autonomic computing paper [2], or domain-specific instances of this architecture that involve pre-defined interactions between small numbers of autonomic element types [8][9][10][11].

What is needed to bridge this gap between vision and reality is a generic technique for the development of systems comprising multiple autonomic elements. The technique should support autonomic elements that can join and leave the system dynamically, which have local objectives to achieve in addition to the overall system objectives, and whose characteristics are unknown at system design time.

This extended version of [1] presents such a generic technique for the development of *autonomic systems of systems* by means of a new type of autonomic computing policy. The new policy type is termed a *resource-definition policy*, and defines the *autonomic system connectors* through which the autonomic manager at the core of an autonomic system should expose the system to its environment.

As illustrated in Figure 1, the implementation of resource-definition policies requires that the reference autonomic computing loop from [2] is augmented with a *synthesise* step. Like in the reference architecture, an *autonomic manager* monitors the system resources through *sensors*, uses its *knowledge* to analyse their state and to plan changes in their configurable parameters, and implements (or "executes") these changes through *effectors*. Additionally, the autonomic manager module that implements the "synthesise" step has the role to dynamically generate connectors that expose the underlying system to its environment as requested by the user-specified resource-definition policies. This new functionality supports the flexible, run-time integration of existing autonomic systems into hierarchical, collaborating or hybrid autonomic systems of systems, as illustrated in Figure 2.

The main contributions of the article are:

1) A theoretical framework for the formal, unified specification and analysis of autonomic computing policies.
2) The introduction of resource-definition policies within this framework.
3) A set of automated code generation techniques for the run-time synthesis of autonomic system connectors as web services, and an augmented version of the author's general-purpose autonomic manager [12][13] that uses these techniques to support the new policy type.

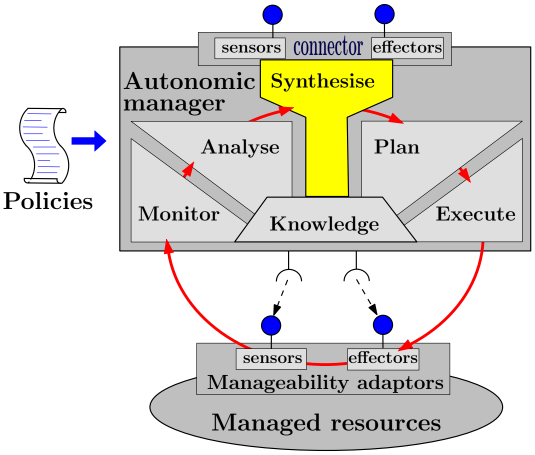


Fig. 1. Policy-based autonomic computing system whose control loop is augmented with a *connector synthesis step*.

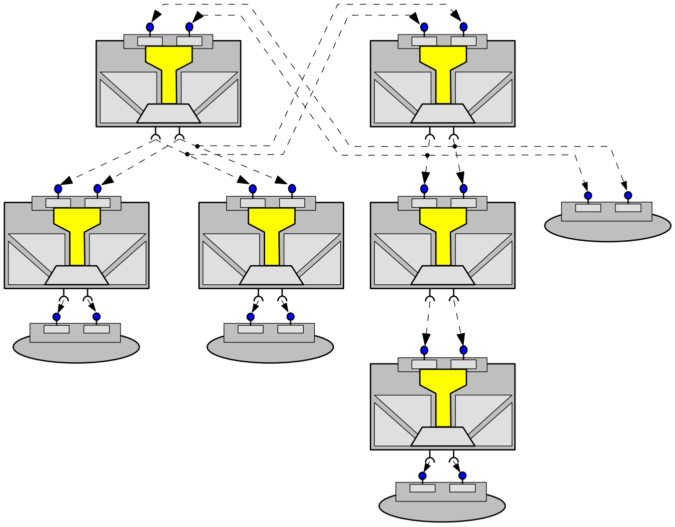


Fig. 2. Resource-definition policies support the run-time integration of autonomic systems into autonomic systems of systems.

4) A case study that illustrates the use of resource-definition policies to implement an autonomic data-centre application.

The remainder of the article starts with an overview of related work in Section II, followed by the description of our framework for the specification of autonomic computing policies in Section III. Section IV provides a formal introduction to resource-definition policies. Section V describes the prototype implementation of an autonomic manager that handles resource-definition policies. System-of-systems application development using the new policy type is discussed in Section VI. Finally, Section VII provides an overview of the problem addressed by our research work and the solution proposed in the article, a summary of the case study used to validate this solution, and a list of future work directions.

II. Related Work

The idea to integrate a set of autonomic systems into a higher-level autonomic system is not new. Indeed, the seminal paper of Kephart and Chess [2] envisaged *autonomic elements* that "interact with other elements and with human programmers via their autonomic managers". More recently, this vision was adapted and extended successfully to specific application domains including aerospace [7] and "beyond 3G" networking [10]. However, these extensions represent work in progress.

The high-level architecture proposed in [7] is also characterised by pre-defined types of interactions between their autonomic elements. Due to the statically implemented sensor-effector interfaces of their autonomic elements, the potential applications of this approach are limited to those pre-planned when the autonomic elements are originally developed.

The ongoing research project whose agenda is described in [10] will employ *event services* to enable flexible interactions among *autonomic network management elements*. When this work will be completed, the functionality of the resulting framework will resemble that of the autonomic system connectors described in our article, but in the area of autonomic networking.

The work described in [9] proposes the use of hierarchical systems of event-driven sensors for monitoring and symptom recognition within autonomic systems. The use of dedicated policies to dynamically configure an instance of the architecture in [9] is somewhat similar to our use of resource-definition policies. However, unlike our approach that supports the automated, run-time synthesis of connectors comprising a combination of autonomic computing sensors and effectors, the framework in [9] focuses exclusively on sensors, and requires manual coding for the implementation of these sensors.

Fully-fledged instances of the data-centre autonomic systems comprising a pair of collaborating autonomic elements and a two-level hierarchy of autonomic elements are described in [8] and [11], respectively. The two successful domain-specific applications by research teams at IBM Research demonstrate the benefits of partitioning the autonomic functionality of large computing systems among multiple autonomic elements. However, the use of hard-coded autonomic manager interfaces limits the inter-operation of the autonomic elements of the applications in [8][11] to pre-defined types of interactions. Furthermore, the need to implement each of these interfaces upfront requires a significant amount of coding effort, and does not scale well to autonomic systems of systems comprising more than a few autonomic elements (the systems in [8] and [11] comprise two and three autonomic managers, respectively).

In contrast with all these approaches, the use of resource-definition policies makes possible the automated, runtime generation of the sensor-effector connectors of an autonomic element. Thus, the main advantage of our novel approach to implementing autonomic systems of systems resides in its unique ability to support the development of unforeseen applications by dynamically (re)defining the interfaces of existing autonomic systems. As an added benefit, the use of resource-definition policies eliminates completely the effort and costs associated with the manual implementation of these interfaces.

The architecture and functionality of the autonomic systems of systems built using resource-definition policies resemble those of intelligent multi-agent systems [14]. However, the standard approaches used for the development of multi-agent systems differ significantly from the approach proposed in this paper. Thus, the typical approach to multi-agent system development [14] involves defining and implementing the agent interfaces statically, before the components of the system are deployed. In contrast, our approach has the advantage that these interfaces can flexibly be defined at runtime, thus enabling the development of applications not envisaged until after the components of the system are deployed.

A unified framework that interrelates three types of autonomic computing policies was introduced in [15]. Based on the concepts of *state* and *action* (i.e., state transition) adopted from the field of artificial intelligence, this framework makes possible the effective high-level analysis of the relationships among action, goal and utility-function policies for autonomic computing. Our policy specification framework differs essentially from the one in [15], which it complements through enabling the formal specification and analysis of all these policy types in terms of the knowledge employed by the autonomic manager that implements them.

III. A Framework for the Formal Specification of Autonomic Computing Policies

Before introducing the new type of policy, we will formally define the knowledge module of the autonomic manager in Figure 1. This knowledge is a tuple that models the $n \geq 1$ resources of the system and their behaviour:

$$K = (R_1, R_2, \ldots, R_n, f), \tag{1}$$

where R_i, $1 \leq i \leq n$ is a formal specification for the ith system resource, and f is a model of the *known* behaviour of the system. Each resource specification R_i represents a named sequence of $m_i \geq 1$ resource parameters, i.e.,

$$R_i = (resId_i, P_{i1}, P_{i2}, \ldots, P_{im_i}), \forall 1 \leq i \leq n, \tag{2}$$

where $resId_i$ is an identifier used to distinguish between different types of resources. Finally, for each $1 \leq i \leq n$, $1 \leq j \leq m_i$, the resource parameter P_{ij} is a tuple

$$P_{ij} = (\mathit{paramId}_{ij}, \mathit{ValueDomain}_{ij}, \mathit{type}_{ij}) \tag{3}$$

where $\mathit{paramId}_{ij}$ is a `string`-valued identifier used to distinguish the different parameters of a resource; $\mathit{ValueDomain}_{ij}$ is the set of possible values for P_{ij}; and $\mathit{type}_{ij} \in \{\texttt{ReadOnly}, \texttt{ReadWrite}\}$ specifies whether the autonomic manager can only read or can both read and modify the value of the parameter. The parameters of each resource must have different identifiers, i.e.,

$$\forall 1 \leq i \leq n \bullet \forall 1 \leq j < k \leq m_i \bullet \mathit{paramId}_{ij} \neq \mathit{paramId}_{ik}$$

We further define the *state space* S of the system as the Cartesian product of the values domains of all its `ReadOnly` resource parameters, i.e.,

$$S = \bigtimes_{1 \leq i \leq n} \; \bigtimes_{\substack{1 \leq j \leq m_i \\ \mathit{type}_{ij} = \texttt{ReadOnly}}} \mathit{ValueDomain}_{ij} \tag{4}$$

Similarly, the *configuration space* C of the system is defined as the Cartesian product of the value domains of all its `ReadWrite` resource parameters, i.e.,

$$C = \bigtimes_{1 \leq i \leq n} \; \bigtimes_{\substack{1 \leq j \leq m_i \\ \mathit{type}_{ij} = \texttt{ReadWrite}}} \mathit{ValueDomain}_{ij} \tag{5}$$

With this notation, the behavioural model f from (1) is a partial function[1]

$$f : S \times C \nrightarrow S$$

such that for any $(\mathbf{s}, \mathbf{c}) \in \mathrm{domain}(f)$, $f(\mathbf{s}, \mathbf{c})$ represents the *expected* future state of the system if its current state is $\mathbf{s} \in S$ and its configuration is set to $\mathbf{c} \in C$. Presenting classes of behavioural models that can support the implementation of different autonomic computing policies is beyond the scope of this paper; for a description of such models see [12].

The standard types of autonomic policies described in [15][11][16] can be defined using this notation as follows:

1) An *action policy* specifies how the system configuration should be changed when the system reaches certain state/configuration combinations:

$$p_{\text{action}} : S \times C \nrightarrow C. \tag{6}$$

A graphical representation of an action policy is shown in Figure 3a. Note that an action policy can be implemented even when the autonomic manager has no knowledge about the behaviour of the managed resources, i.e., when $\mathrm{domain(f)} = \emptyset$ in eq. (1).

2) A *goal policy* partitions the state/configuration combinations for the system into desirable and undesirable:

$$p_{\text{goal}} : S \times C \rightarrow \{\texttt{true}, \texttt{false}\}, \tag{7}$$

with the autonomic manager requested to maintain the system in an operation area for which p_{goal} is `true`. Figure 3b shows a graphical representation of a goal policy within the theoretical framework introduced in this section.

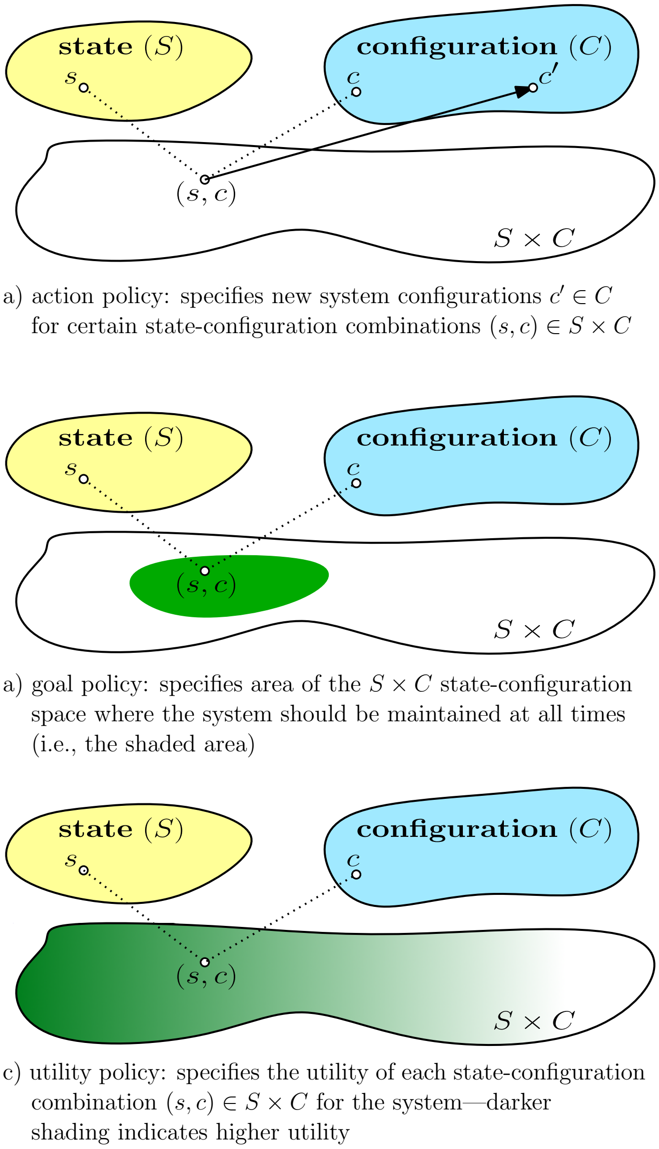
3) A *utility policy* (Figure 3c) associates a value with each state/configuration combination, and the autonomic manager should adjust the system configuration such as to maximise this value:

$$p_{\text{utility}} : S \times C \rightarrow \mathbb{R}. \tag{8}$$

Example 1 To illustrate the application of the notation introduced so far, consider the example of an autonomic data-centre comprising a pool of $nServers \geq 0$ servers that need to be partitioned among the $N \geq 1$ services that the data-centre can provide. Assume that each such service has a $priority$ and is subjected to a variable $workload$. The knowledge (1) for this system can be expressed as a tuple

$$K = (\mathit{ServerPool}, \mathit{Service}_1, \ldots, \mathit{Service}_n, f) \tag{9}$$

[1] A partial function on a set X is a function whose domain is a subset of X. We use the symbol $\nrightarrow$ to denote partial functions.



a) action policy: specifies new system configurations $c' \in C$ for certain state-configuration combinations $(s,c) \in S \times C$

a) goal policy: specifies area of the $S \times C$ state-configuration space where the system should be maintained at all times (i.e., the shaded area)

c) utility policy: specifies the utility of each state-configuration combination $(s,c) \in S \times C$ for the system—darker shading indicates higher utility

Fig. 3. Graphical representation of the standard types of autonomic computing policies

where the models for the server pool and for a generic service i, $1 \leq i \leq N$ are given by:

$$\begin{aligned} ServerPool = (&\texttt{"serverPool"}, \\ &(\texttt{"nServers"}, \mathbb{N}, \texttt{ReadOnly}), \\ &(\texttt{"partition"}, \mathbb{N}^N, \texttt{ReadWrite})) \\ Service_i = (&\texttt{"service"}, \\ &(\texttt{"priority"}, \mathbb{N}_+, \texttt{ReadOnly}), \\ &(\texttt{"workload"}, \mathbb{N}, \texttt{ReadOnly})) \end{aligned} \tag{10}$$

The state and configuration spaces of the system are $S = \mathbb{N} \times (\mathbb{N}_+ \times \mathbb{N})^N$ and $C = \mathbb{N}^N$, respectively. For simplicity, we will consider that the *workload* of a service represents the minimum number of operational servers it requires to achieve its service-level agreement. Sample action, goal and utility policies for the system are specified below by giving their values for a generic data-centre state $s = (n, p_1, w_1, p_2, w_2, \ldots, p_N, w_N) \in S$ and configuration $c = (n_1, n_2, \ldots, n_N) \in C$:

$$p_{\text{action}}(s,c) = (\lceil \alpha w_1 \rceil, \lceil \alpha w_2 \rceil, \ldots, \lceil \alpha w_N \rceil) \tag{11}$$

$$\begin{aligned} &p_{\text{goal}}(s,c) = \forall 1 \leq i \leq N \bullet \\ &\quad (n_i > 0 \implies (\forall 1 \leq j \leq N \bullet p_j > p_i \implies n_j = \lceil \alpha w_j \rceil)) \wedge \\ &\quad \left(n_i = 0 \implies \sum_{\substack{j=1 \\ p_j \geq p_i}}^{N} n_j = n \right) \end{aligned} \tag{12}$$

$$p_{\text{utility}}(s,c) = \begin{cases} -\infty, & \text{if } \sum_{i=1}^{N} n_i > n \\ \sum_{\substack{i=1 \\ w_i > 0}}^{N} p_i u(w_i, n_i) - \epsilon \sum_{i=1}^{N} n_i, & \text{otherwise} \end{cases} \tag{13}$$

We will describe these policies one at a time. First, the action policy (11) prescribes that $\lceil \alpha w_i \rceil$ servers are allocated to service i, $1 \leq i \leq N$ at all times. Notice how a *redundancy factor* $\alpha \in (1,2)$ is used in a deliberately simplistic attempt to increase the likelihood that at least w_i servers will be available for service i in the presence of server failures. Also, the policy is (over)optimistically assuming that $n \geq \sum_{i=1}^{N} \lceil \alpha w_i \rceil$ at all times.

The goal policy (12) specifies that the desirable state/configuration combinations of the data-centre are those in which two conditions hold for each service i, $1 \leq i \leq N$. The first of these conditions states that a service should be allocated servers only if all services of higher priority have already been allocated $\lceil \alpha w_i \rceil$ servers. The second condition states that a service should be allocated no server only if all n available servers have been allocated to services of higher or equal priority.

Finally, the utility policy requires that the value of the expression in (13) is maximised. The value $-\infty$ in this expression is used to avoid the configurations in which more servers than the n available are allocated to the services. When this is not the case, the value of the policy is given by the combination of two sums. The first sum encodes the utility $u(w_i, n_i)$ of allocating n_i servers to each service i with $w_i > 0$, weighted by the priority p_i of the service. By setting ϵ to a small positive value (i.e., $0 < \epsilon \ll 1$), the second sum ensures that from all server partitions that maximise the first sum, the one that uses the smallest number of servers is chosen at all times. A sample function u is shown in Figure 4. More realistic utility functions u and matching behavioural models f from eq. (9) are described in [12].

IV. Resource-Definition Policies

A. Definition

Using $\mathcal{R}$ to denote the set of all resource models with the form in (2), and $\mathcal{E}(S,C)$ to denote the set of all expressions defined on the Cartesian product $S \times C$, we can now give the generic form of a resource-definition policy as

$$p_{\text{def}} : S \times C \to \mathcal{R} \times \mathcal{E}(S,C)^q, \tag{14}$$

$$
\begin{aligned}
&u : \mathsf{R}_+ \times \mathsf{R}_+ \to [0,1]\\
&u(w,n) = \begin{cases} 0, & \text{if } n < (2-\alpha)w \\ \frac{n-(2-\alpha)w}{2(\alpha-1)w}, & \text{if } (2-\alpha)w \le n \le \alpha w \\ 1, & \text{if } n > \alpha w \end{cases}\\
&\alpha \in (1,2)
\end{aligned}
$$

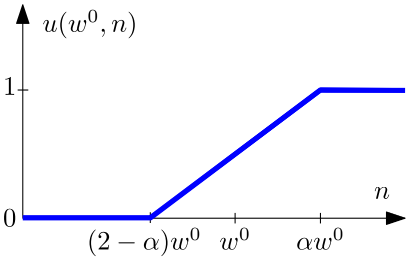


Fig. 4. Sample function u for Example 1—the graph shows u for a fixed value of its first argument, i.e., $w = w^0$

where, for any $(\mathbf{s}, \mathbf{c}) \in S \times C$,

$$p_{\text{def}}(\mathbf{s}, \mathbf{c}) = (R, E_1, E_2, \ldots, E_q). \tag{15}$$

In this definition, R represents the resource that the autonomic manager is required to synthesise, and the expressions E_1, E_2, $\ldots$, E_q specify how the autonomic manager will calculate the values of the $q \geq 0$ `ReadOnly` parameters of R as functions of $(\mathbf{s}, \mathbf{c})$. Assuming that the value domain for the ith `ReadOnly` parameter of R, $1 \leq i \leq q$ is $\mathit{ValueDomain}_i^{\text{synth\_RO}}$, we have

$$E_i : S \times C \to \mathit{ValueDomain}_i^{\text{synth\_RO}}. \tag{16}$$

Example 2 Consider again the autonomic data-centre from Example 1. A sample resource-definition policy that complements the utility policy in (13) is given by

$$
\begin{aligned}
p_{\text{def}}(s,c) = ((&\texttt{"dataCentre"},\\
&(\texttt{"id"}, \mathit{String}, \texttt{ReadOnly})\\
&(\texttt{"maxUtility"}, \mathbb{R}, \texttt{ReadOnly}),\\
&(\texttt{"utility"}, \mathbb{R}, \texttt{ReadOnly})),\\
&\texttt{"dataCentre A"},\\
&\max_{(x_1,x_2,\ldots,x_N)\in\mathbb{N}^N} \textstyle\sum_{w_i>0}^{1\leq i\leq N} p_i u(w_i, x_i),\\
&\textstyle\sum_{w_i>0}^{1\leq i\leq N} p_i u(w_i, n_i))
\end{aligned} \tag{17}
$$

This policy requests the autonomic manager to synthesise a resource termed a `"dataCentre"`. This resource comprises three `ReadOnly` parameters: `id` is a string-valued identifier with the constant value `"dataCentre A"`, while `maxUtility` and `utility` represent the maximum and actual utility values associated with the autonomic data-centre when it implements the utility policy (13). (The term $\epsilon \sum_{i=1}^{N} n_i$ from the policy definition is insignificant, and was not included in (17) for simplicity.) Exposing the system through this synthesised resource enables an external autonomic manager to monitor how close the data-centre is to achieving its maximum utility.

Note that the generic form of a resource-definition policy (14)-(15) allows the policy to request the autonomic manager to synthesise different types of resources for different state/configuration combinations of the system. While the preliminary use cases that we have studied so far can be handled using resource-definition policies in which the resource model R from (15) is fixed for all $(\mathbf{s}, \mathbf{c}) \in S \times C$, we envisage that this capability will be useful for more complex applications of resource-definition policies.

B. *Synthesised Resources with* `ReadWrite` *Parameters*

In this section we explore the semantics and applications of `ReadWrite` (i.e., configurable) parameters in synthesised resources. These are parameters whose identifiers and value domains are specified through a resource-definition policy, but whose values are set by an external entity such as another autonomic manager. Because these parameters do not correspond to any element of the managed resources within the autonomic system, the only way to ensure that they have an influence on the autonomic system in Figure 1 as a whole is to take them into account within the set of policies implemented by the autonomic manager. This is achieved by redefining the state space S of the system. Thus, in the presence of resource-definition policies requesting the synthesis of high-level resources with a non-empty set of `ReadWrite` parameters $\{P_1^{\text{synth\_RW}}, P_2^{\text{synth\_RW}}, \ldots, P_r^{\text{synth\_RW}}\}$, the state space definition (4) is replaced by:

$$
\begin{aligned}
S = &\left(\bigtimes_{1\leq i\leq n} \bigtimes_{\substack{1\leq j\leq m_i\\ \mathit{type}_{ij}=\texttt{ReadOnly}}} \mathit{ValueDomain}_{ij} \right) \times\\
&\left(\bigtimes_{1\leq i\leq r} \mathit{ValueDomain}_i^{\text{synth\_RW}} \right),
\end{aligned} \tag{18}
$$

where $\mathit{ValueDomain}_i^{\text{synth\_RW}}$ represents the value domain of the ith synthesised resource parameter P_i^{synth}, $1 \leq i \leq r$. Given that the synthesised sensors involve setting the value of the connector parameters by the autonomic manager, the configuration space of the system can be redefined in an analogous way as:

$$
\begin{aligned}
C = &\left(\bigtimes_{1\leq i\leq n} \bigtimes_{\substack{1\leq j\leq m_i\\ \mathit{type}_{ij}=\texttt{ReadWrite}}} \mathit{ValueDomain}_{ij} \right) \times\\
&\left(\bigtimes_{1\leq i\leq q} \mathit{ValueDomain}_i^{\text{synth\_RO}} \right).
\end{aligned} \tag{19}
$$

Figure 5 illustrates graphically the way in which a policy-definition policy extends the state and configuration spaces of an autonomic system.

Example 3 Consider again our running example of an autonomic data-centre. The resource-definition policy in (17) can be extended to allow a peer data-centre (such as a data-centre

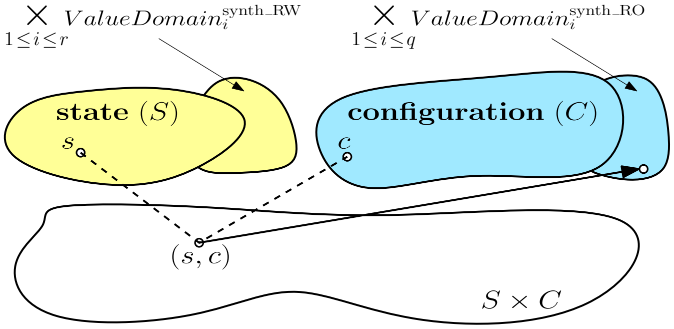

$\times_{1\leq i\leq r} ValueDomain_i^{\text{synth\_RW}}$
$\times_{1\leq i\leq q} ValueDomain_i^{\text{synth\_RO}}$
state (S)
s
configuration (C)
c
(s, c)
$S \times C$

Fig. 5. Resource-definition policies extend the state and configuration spaces of an autonomic system with the value domains of the synthesised connectors.

running the same set of services within the same security domain) to take advantage of any spare servers:

$$
\begin{aligned}
p'_{\text{def}}(s,c) = ((&\texttt{"dataCentre"},\\
&(\texttt{"id"}, \mathit{String}, \texttt{ReadOnly})\\
&(\texttt{"maxUtility"}, \mathbb{R}, \texttt{ReadOnly}),\\
&(\texttt{"utility"}, \mathbb{R}, \texttt{ReadOnly})),\\
&(\texttt{"nSpare"}, \mathbb{N}, \texttt{ReadOnly})),\\
&(\texttt{"peerRequest"}, \mathbb{N}^N, \texttt{ReadWrite})),\\
&\texttt{"dataCentre A"},\\
&\max_{(x_1,x_2,\ldots,x_N)\in\mathbb{N}^N} \textstyle\sum_{w_i>0}^{1\leq i\leq N} p_i u(w_i, x_i),\\
&\textstyle\sum_{w_i>0}^{1\leq i\leq N} p_i u(w_i, n_i),\ n - \sum_{i=1}^{N} n_i)
\end{aligned}
\tag{20}
$$

The synthesised resource has two new parameters: `nSpare` represents the number of servers not allocated to any (local) service; and `peerRequest` is a vector $(n_1^l, n_2^l, \ldots, n_N^l)$ that a remote data-centre can set to request that the local data-centre assigns n_i^l of its servers to service i, for all $1 \leq i \leq N$.

To illustrate how this is achieved, we will consider two data-centres that each implements the policy in (20), and which have access to each other's "dataCentre" resource as shown in the lower half of Figure 10. For simplicity, we will further assume that the data-centres are responsible for disjoint sets of services (i.e., there is no $1 \leq i \leq N$ such that $w_i > 0$ for both data-centres). To ensure that the two data-centres collaborate, we need policies that specify how each of them should set the $\texttt{peerRequest}^r$ parameter of its peer, and how it should use its own $\texttt{peerRequest}^l$ parameter (which is set by the other data-centre). The "dataCentre" parameters have been annotated with ^{l} and ^{r} to distinguish between identically named parameters belonging to the local and remote data-centre, respectively. Before giving a utility policy that ensures the collaboration of the two data-centres, it is worth mentioning that the state of each has the form $s = (n, p_1, w_1, p_2, w_2, \ldots, p_N, w_N, n^r, n_1^l, n_2^l, \ldots, n_N^l)$ (cf. (18)); and the system configuration has the form $c = (n_1, n_2, \ldots, n_N, n_1^r, n_2^r, \ldots, n_N^r)$. The utility policy to use alongside policy (20) is given below:

$$
p'_{\text{utility}}(s,c) =
$$
$$
= \begin{cases}
-\infty, & \text{if } \sum\limits_{i=1}^{N} n_i > n \vee \sum\limits_{i=1}^{N} n_i^r > n^r \\
\begin{aligned}&\sum\limits_{\substack{i=1\\w_i>0}}^{N} p_i u(w_i, n_i + n_i^r) - \epsilon \sum\limits_{i=1}^{N} n_i - \\ &-\lambda \sum\limits_{i=1}^{N} n_i^r + \mu \sum\limits_{\substack{i=1\\n_i^l>0}}^{N} \min\left(1, \tfrac{n_i}{n_i^l}\right)\end{aligned} & \text{, otherwise}
\end{cases}
\tag{21}
$$

where $0 < \epsilon \ll \lambda, \mu \ll 1$ are user-specified constants. The value $-\infty$ is used to avoid the configurations in which more servers than available (either locally or from the remote data-centre) are allocated to the local services. The first two sums in the expression that handles all other scenarios are similar to those from utility policy (13), except that $n_i + n_i^r$ rather than n_i servers are being allocated to any local service i for which $w_i > 0$. The term $-\lambda \sum_{i=1}^{N} n_i^r$ ensures that the optimal utility is achieved with as few remote servers as possible, and the term $\mu \sum_{n_i^l>0}^{1\leq i\leq N} \min(1, \frac{n_i}{n_i^l})$ requests the policy engine to allocate local servers to services for which $n_i^l > 0$. Observe that the contribution of a term $\mu \min(1, \frac{n_i}{n_i^l})$ to the overall utility increases as n_i grows from 0 to n_i^l, and stays constant if n_i increases beyond n_i^l. Together with the utility term $-\epsilon \sum_{i=1}^{N} n_i$, this determines the policy engine to never allocate more than the requested n_i^l servers to service i. Small positive constants are used for the weights ϵ, λ and μ so that the terms they belong to are negligible compared to the first utility term. Further, choosing $\epsilon \ll \lambda$ ensures that using a local server decreases the utility less than using a remote one; and setting $\epsilon \ll \mu$ ensures that allocating up to n_i^l servers to a service i at the request of the remote data-centre increases the system utility.

Finally, note that because the requests for remote servers and the allocation of such servers take place asynchronously, there is a risk that the parameter values used in policy (21) may be out of date.[2] However, this is not a problem, as the allocation of fewer or more remote servers than ideally required is never decreasing the utility value for a data-centre below the value achieved when the data-centre operates in isolation. Additionally, local servers are never used for remote services at the expense of the local services because $\sum_{w_i>0}^{1\leq i\leq N} p_i u(w_i, n_i) \gg \mu \sum_{n_i^l>0}^{1\leq i\leq N} \min(1, n_i/n_i^l))$ in the utility expression.

V. Prototype Implementation

The *policy engine* (i.e., policy-based autonomic manager) introduced by the author in [13] was extended with the ability to handle the new type of autonomic computing policy. Implemented as a model-driven, service-oriented architecture with the characteristics presented in [12], the policy engine

[2] In practical scenarios that we investigated this happened very infrequently relative to the time required to solve the linear optimisation problem (21) automatically within the policy engine.

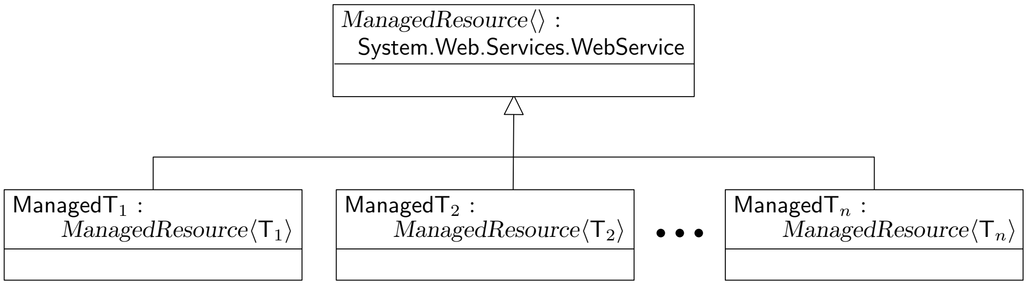

ManagedResource⟨⟩ : System.Web.Services.WebService
ManagedT$_1$: ManagedResource⟨T$_1$⟩
ManagedT$_2$: ManagedResource⟨T$_2$⟩
• • •
ManagedT$_n$: ManagedResource⟨T$_n$⟩

Fig. 6. Generic programming is used to implement the manageability adaptors for the system resources.

from [13] can manage IT resources whose model is supplied to the engine in a runtime configuration step. The IT resource models described by eq. (1) are represented as XML documents that are instances of a pre-defined meta-model encoded as an XML schema [12][13]. This choice was motivated by the availability of numerous off-the-shelf tools for the manipulation of XML documents and XML schemas—a characteristic largely lacking for the other technologies we considered. The policy engine is implemented as a .NET web service. In its handling of IT resources whose characteristics are unknown until runtime, the policy engine takes advantage of object-oriented (OO) technology features including:

- *polymorphism*—the ability to use instances of derived data types in place of instances of their base data types;
- *reflection*—an OO programming technique that allows the runtime discovery and creation of objects based on their metadata [17].
- *generic programming*—the ability to write code in terms of data types unknown until runtime [18].

The manageability adaptors from Figure 1 are implemented by the framework in [13] as web services that specialise a generic, abstract web service *ManagedResource*⟨⟩. For each type of resource in the system, a manageability adaptor is built in two steps. First, a class (i.e., a data type) T_i is generated from the resource model (2) that will be used to configure the policy engine, $\forall 1 \leq i \leq n$. Second, the manageability adaptor $\mathsf{ManagedT}_i$ for resources of type T_i is implemented by specialising our generic *ManagedResource*⟨⟩ adaptor, i.e., $\mathsf{ManagedT}_i$: *ManagedResource*⟨T_i⟩. This process is illustrated in Figure 6 and described in detail in [13].

Adding support for the implementation of the resource-definition policy in (14)–(15) required the extension of the policy engine described above with the following functionality:

1) The automated generation of a .NET class T for the synthesised resource R from (15). This class is built by including a field and the associated getter/setter methods for each parameter of R. The types of these fields are given by the value domains of the resource parameters.
2) The automated creation of an instance of T. Reflection is employed to create an instance of T for the lifespan of the resource-definition policy. The `ReadOnly` fields of this object are updated by the policy engine using the expressions E_1, E_2, …, E_q from eq. (16); the updates are carried out whenever the object is accessed by an external entity.
3) The automated generation of a manageability adaptor web service ManagedT : *ManagedResource*⟨T⟩. The web methods provided by this *autonomic system connector* allow entities from outside the autonomic system (e.g., external autonomic managers) to access the object of type T maintained by the policy engine. The fields of this object that correspond to `ReadOnly` parameters of R can be read, and those corresponding to `ReadWrite` parameters can be read and modified, respectively.

The .NET components generated in steps 1 and 3 are deployed automatically, and made accessible through the same Microsoft IIS instance as the policy engine. The synthesised connector is available as soon as the engine completes its handling of the resource-definition policy. The URL of the connector has the same prefix as that of the policy engine, and ends in "Managed$resID$.asmx", where $resID$ is the resource identifier specified in the resource-definition policy. For instance, if the policy engine URL is `http://www.abc.org/PolicyEngine.asmx`, the synthesised connector for a resource of type dataCentre will be accessible at the address `http://www.abc.org/ManagedDataCentre.asmx`.

Example 4 Returning to our running example of an autonomic data-centre, the class diagram in Figure 7 depicts the manageability adaptors in place after policy (20) was supplied to the policy engine. Thus, the ManagedServerPool and ManagedService classes in this diagram represent the manageability adaptors implemented manually for the $ServerPool$ and $Service$ resources described in Example 1. The other manageability adaptor derived from *ManagedResource*⟨⟩ (i.e., ManagedDataCentre) was synthesised automatically by the policy engine as a result of handling the resource-definition policy.

Also shown in the diagram are the classes used to represent instances of the IT resources within the system—serverPool and service for the original autonomic system, and dataCentre for the resource synthesised from policy (20). Notice the one-to-one mapping between the fields of these classes and the parameters of their associated resources (described in Examples 1 and 3).

VI. Application Development

System-of-systems application development with resource-definition policies involves supplying such policies to existing

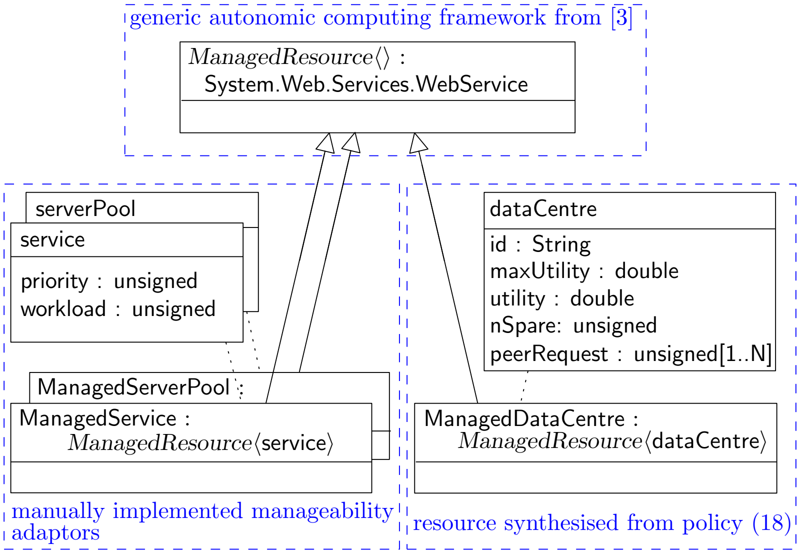

generic autonomic computing framework from [3]
ManagedResource⟨⟩ : System.Web.Services.WebService
serverPool
service
priority : unsigned
workload : unsigned
ManagedServerPool :
ManagedService : ManagedResource⟨service⟩
dataCentre
id : String
maxUtility : double
utility : double
nSpare: unsigned
peerRequest : unsigned[1..N]
ManagedDataCentre : ManagedResource⟨dataCentre⟩
manually implemented manageability adaptors
resource synthesised from policy (18)

Fig. 7. Class diagram for Example 4

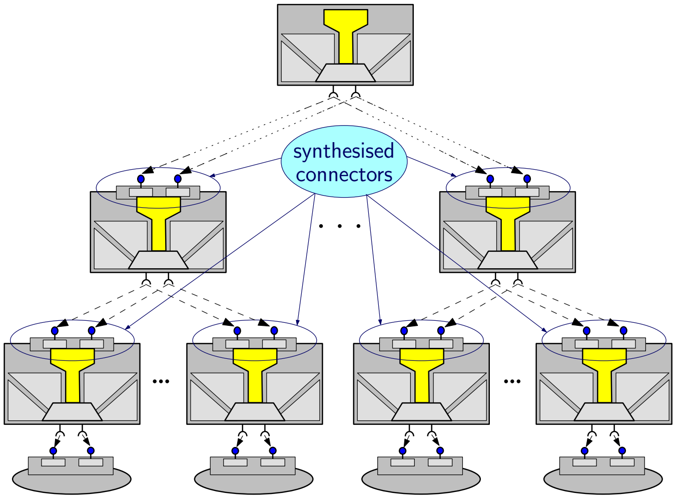


Fig. 8. Hierarchical system-of-systems development with resource-definition policies

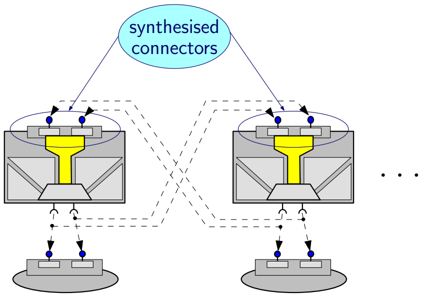


Fig. 9. Collaborating autonomic systems that inter-operate by means of run-time synthesised connectors

autonomic systems whose autonomic managers support the new policy type. Hierarchical systems of systems can be built by setting a higher-level autonomic manager to monitor and/or control subordinate autonomic systems through synthesised connectors that abstract out the low-level details of the systems they expose to the outside world. As illustrated in Figure 8, the hierarchy of autonomic systems can comprise any number of levels.

Alternatively, the original autonomic systems can be configured to collaborate with each other by means of the synthesised resource sensors and effectors. In this scenario, the sensors are used to expose the global state of each system and the effectors are employed to provide *services* to peer autonomic systems as envisaged in the seminal paper of Kephart and Chess [2]. The approach is illustrated in Figure 9.

Hybrid applications comprising both types of interactions mentioned above are also possible, as illustrated by the following example.

Example 5 The policy engine from Section V was used to simulate an autonomic system of systems comprising the pair of autonomic data-centres described in Example 3, and a top-level autonomic manager that monitors and summarises their performance using a dashboard resource (Figure 10). The policies implemented by the autonomic managers local to each data-centre are policies (20)–(21) from Example 3. The top-level autonomic manager implements a simple action policy that periodically copies the values of the `maxUtility` and `utility` parameters of the `"dataCentre"` resources synthesised by the data-centres into the appropriate `ReadWrite` parameters of the dashboard. For brevity, we do not give this policy here; a sample action policy was presented earlier in Example 1.

We used the data-centre resource simulators from [12], and implemented the dashboard resource as an ASP.NET web page provided with a manageability adaptor built manually as described in Section V and in [13]. Separate series of experiments for 20-day simulated time periods we run for two scenarios. In the first scenario, the data-centres were kept operating in isolation, by blocking the mechanisms they could use to discover each other. In the second scenario, the

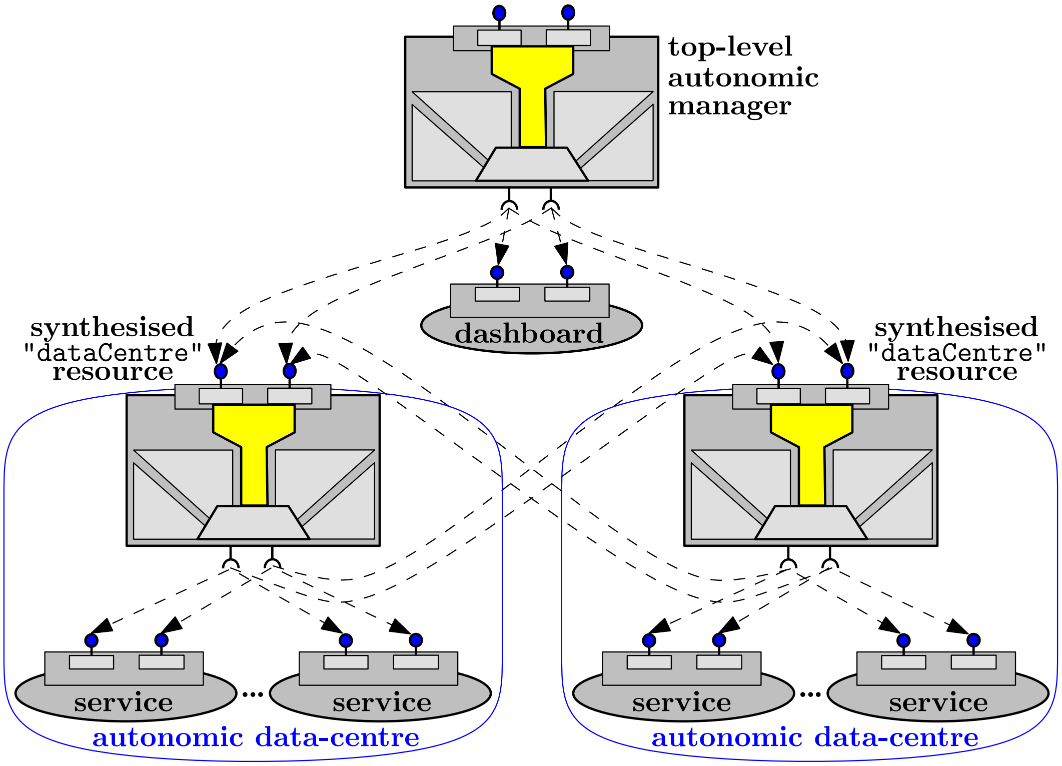


Fig. 10. Autonomic system of systems

data-centres were allowed to discover each other, and thus to collaborate through implementing policy (21). Figure 11 depicts typical snapshots of the dashboard for both scenarios and for one of the data-centres; the same simulated service workloads were used in both experiments shown. As expected from the analysis in Example 3, the system achieves higher utility when data-centre collaboration is enabled, thus allowing data-centres to utilise each other's spare servers.

VII. Conclusion and Future Work

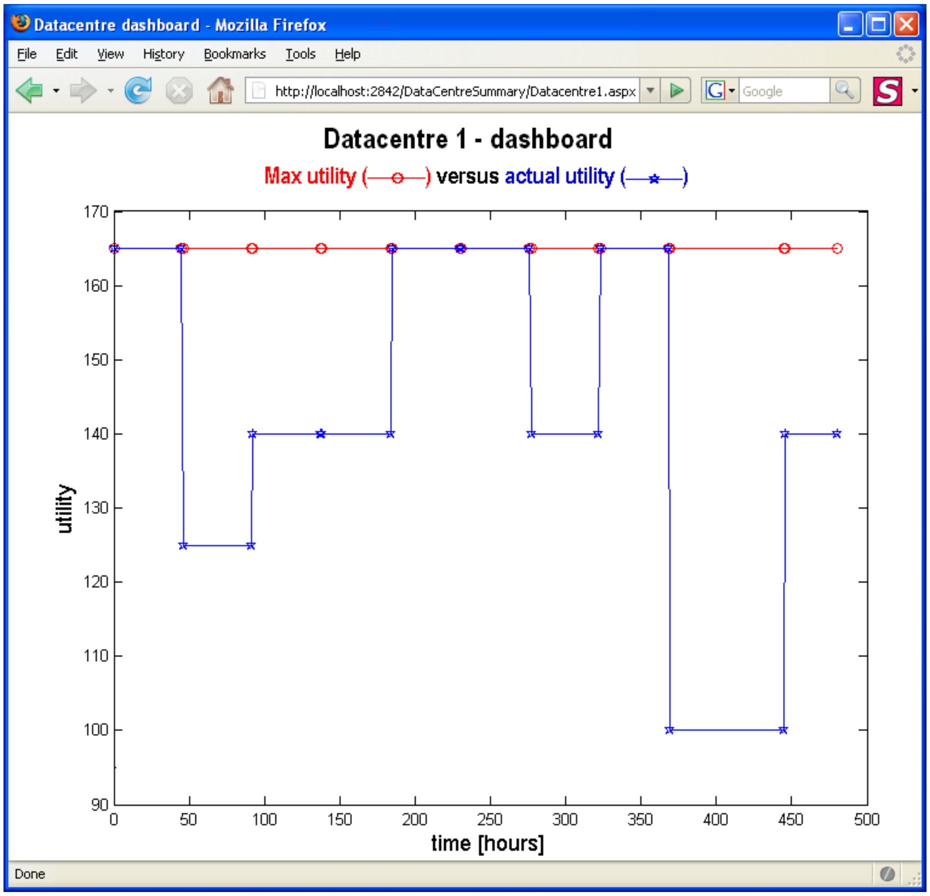
Many of today's computing systems are built through the integration of components that are systems in their own right: they operate autonomously and follow local goals in addition to contributing to the realisation of the global objectives of the overall system. The research results presented in this article enable the addition of self-managing capabilities to this class of systems. This is achieved through the run-time generation of *autonomic system connectors* that can be used to dynamically assemble autonomic systems of systems out of a set of individual autonomic systems.

We introduced a policy specification framework for autonomic computing, and used it to formally define the existing types of autonomic computing policies from [15][11][16] and a new type of policy that we termed a resource-definition policy. We described the semantics of resource-definition policies, and explained how they can be used to dynamically generate connectors that enable the integration of multiple autonomic systems into autonomic systems of systems. This represents a significant step towards addressing a major challenge of autonomic computing, namely the seamless integration of multiple autonomic elements into larger self-managing computing systems [2].

Additionally, the article presented how resource-definition policies can be implemented by augmenting the reference autonomic computing loop with a *connector synthesis* step, and described a prototype implementation of an autonomic manager that comprises this additional step.

To validate the proposed approach to developing autonomic systems of systems, we used the prototype autonomic manager in a case study that was presented as a running example in Sections III to VI. The case study involved the adaptive allocation of data-centre servers to services with different priorities and variable workloads. The system of systems considered by our case study included a couple of simulated autonomic data-centres that employed utility-function policies to dynamically partition their physical servers among local services. Within this autonomic system of systems, resource-definition policies were used successfully to synthesise a `"dataCentre"` resource fulfilling two distinct purposes:

1) Collaboration between the autonomic data-centres. Each data-centre exposed its number of unused, spare servers through a `ReadOnly` parameter of the synthesised resource, and the peer data-centre could set a `ReadWrite` parameter of the same resource to request that some of its services are run on these spare servers. The effectiveness of the technique was demonstrated by the improved utility achieved by the simulated data-centres when this collaboration was enabled, compared to the scenario when the data-centres operated in isolation from each other.


Datacentre dashboard - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:2842/DataCentreSummary/Datacentre1.aspx
Google
Datacentre 1 - dashboard
Max utility (—o—) versus actual utility (—*—)
utility
170
160
150
140
130
120
110
100
90
0
50
100
150
200
250
300
350
400
450
500
time [hours]
Done

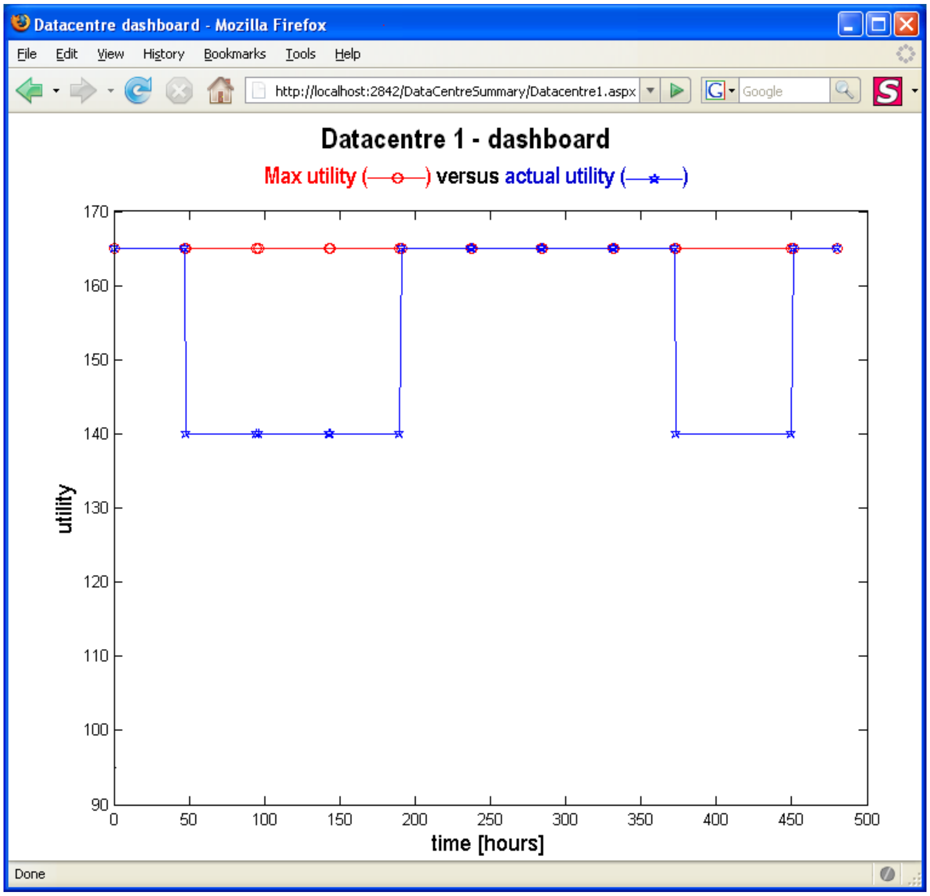

Datacentre dashboard - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://localhost:2842/DataCentreSummary/Datacentre1.aspx
Google
Datacentre 1 - dashboard
Max utility (—o—) versus actual utility (—*—)
utility
170
160
150
140
130
120
110
100
90
0
50
100
150
200
250
300
350
400
450
500
time [hours]
Done

Fig. 11. Dashboard for isolated data-centre (top) and for identical data-centre operating as part of the autonomic system of systems from Figure 10 (bottom).

2) High-level reporting of data-centre utility. Two `ReadOnly` parameters of the synthesised resource were used to reflect the instantaneous data-centre utility and maximum (or ideal) utility attainable, respectively. A top-level instance of the prototype autonomic manager was employed to monitor this information and, through suitable action policies, to forward it to a web-based dashboard for real-time plotting.

The two uses of resource-definition policies show that both federations of collaborative autonomic elements and hierarchical autonomic systems can be implemented successfully using the tool-supported development technique proposed in the article.

The work presented in the article is part of an ongoing project aimed at devising novel approaches for the effective development of autonomic systems and systems of systems. Related future objectives of this project include extending the policy specification framework with techniques for the analysis of policy properties such as state/configuration space coverage and conflict freeness, and applying resource-definition policies to additional, real-world case studies from the area of data-centre resource management.

Acknowledgement

This work was partly supported by the UK EPSRC grant EP/F001096/1.

References

[1] R. Calinescu, "Resource-definition policies for autonomic computing," in *Proceedings of the 5th International Conference on Autonomic and Autonomous Systems (ICAS 2009)*, 2009, pp. 111–116.

[2] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *IEEE Computer Journal*, vol. 36, no. 1, pp. 41–50, January 2003.

[3] M. Huebscher and J. McCann, "A survey of autonomic computing—degrees, models, and applications," *ACM Comput. Surv.*, vol. 40, no. 3, pp. 1–28, 2008.

[4] M. Parashar and S. Hariri, *Autonomic Computing: Concepts, Infrastructure & Applications*. CRC Press, 2006.

[5] Carnegie Mellon Software Engineering Institute, *Ultra-Large-Scale Systems. The Software Challenge of the Future*. Carnegie Mellon University, 2006.

[6] G. Goth, "Ultralarge systems: Redefining software engineering?" *IEEE Software*, vol. 25, no. 3, pp. 91–94, May/June 2008.

[7] M. Hinchey *et al.*, "Modeling for NASA autonomous nano-technology swarm missions and model-driven autonomic computing," in *Proc. 21st Intl. Conf. Advanced Networking and Applications*, 2007, pp. 250–257.

[8] J. O. Kephart, H. Chan, R. Das, D. W. Levine, G. Tesauro, F. L. R. III, and C. Lefurgy, "Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs," in *Proceedings Fourth International Conference on Autonomic Computing (ICAC'07)*, 2007.

[9] J. Li, P. Martin, W. Powley, K. Wilson, and C. Craddock, "A sensor-based approach to symptom recognition for autonomic systems," in *Proceedings of the 5th International Conference on Autonomic and Autonomous Systems (ICAS 2009)*, 2009, pp. 45–50.

[10] D. Raymer *et al.*, "From autonomic computing to autonomic networking: an architectural perspective," in *Proc. 5th IEEE Workshop on Engineering of Autonomic and Autonomous Systems*, 2008, pp. 174–183.

[11] W. Walsh *et al.*, "Utility functions in autonomic systems," in *Proc. 1st Intl. Conf. Autonomic Computing*, 2004, pp. 70–77.

[12] R. Calinescu, "General-purpose autonomic computing," in *Autonomic Computing and Networking*, M. Denko *et al.*, Eds. Springer, 2009, pp. 3–20.

[13] ——, "Implementation of a generic autonomic framework," in *Proc. 4th Intl. Conf. Autonomic and Autonomous Systems*, D. Greenwood *et al.*, Eds., March 2008, pp. 124–129.

[14] M. Wooldridge, *An Introduction to Multi-agent Systems*. J. Wiley and Sons, 2002.

[15] J. O. Kephart and W. E. Walsh, "An artificial intelligence perspective on autonomic computing policies," in *Proc. 5th IEEE Intl. Workshop on Policies for Distributed Systems and Networks*, 2004.

[16] S. White *et al.*, "An architectural approach to autonomic computing," in *Proc. 1st IEEE Intl. Conf. Autonomic Computing*. IEEE Computer Society, 2004, pp. 2–9.

[17] J. M. Sobel and D. P. Friedman, "An introduction to reflection-oriented programming," in *In Proceedings of Reflection96*, 1996.

[18] R. Garcia *et al.*, "A comparative study of language support for generic programming," *ACM SIGPLAN Notices*, vol. 38, no. 11, pp. 115–134, November 2003.

iaria journals

www.iariajournals.org

International Journal On Advances in Intelligent Systems

- ICAS, ACHI, ICCGI, UBICOMM, ADVCOMP, CENTRIC, GEOProcessing, SEMAPRO, BIOSYSCOM, BIOINFO, BIOTECHNO, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS
- issn: 1942-2679

International Journal On Advances in Internet Technology

- ICDS, ICIW, CTRQ, UBICOMM, ICSNC, AFIN, INTERNET, AP2PS, EMERGING
- issn: 1942-2652

International Journal On Advances in Life Sciences

- eTELEMED, eKNOW, eL&mL, BIODIV, BIOENVIRONMENT, BIOGREEN, BIOSYSCOM, BIOINFO, BIOTECHNO
- issn: 1942-2660

International Journal On Advances in Networks and Services

- ICN, ICNS, ICIW, ICWMC, SENSORCOMM, MESH, CENTRIC, MMEDIA, SERVICE COMPUTATION
- issn: 1942-2644

International Journal On Advances in Security

- ICQNM, SECURWARE, MESH, DEPEND, INTERNET, CYBERLAWS
- issn: 1942-2636

International Journal On Advances in Software

- ICSEA, ICCGI, ADVCOMP, GEOProcessing, DBKDA, INTENSIVE, VALID, SIMUL, FUTURE COMPUTING, SERVICE COMPUTATION, COGNITIVE, ADAPTIVE, CONTENT, PATTERNS
- issn: 1942-2628

International Journal On Advances in Systems and Measurements

- ICQNM, ICONS, ICIMP, SENSORCOMM, CENICS, VALID, SIMUL
- issn: 1942-261x

International Journal On Advances in Telecommunications

- AICT, ICDT, ICWMC, ICSNC, CTRQ, SPACOMM, MMEDIA
- issn: 1942-2601