Proceedings of the 8th IEEE GCC Conference and Exhibition, Muscat, Oman, 1-4 February, 2015

Smart Buildings and the Human-Machine Cloud

Nikolaos Mavridis
NCSR Demokritos, Greece
Email: nmavridis@iit.demokritos.gr

Georgios Pierris
University of South Wales, UK
Email: georgios.pierris@students.southwales.ac.uk

Chiraz BenAbdelkader
University of Manouba, Republic of Tunisia
Email: chiraz.benabdelkader@gmail.com

Aleksandar Krstikj
University of Southern California, USA
Email: krstikj@usc.edu

Christos Karaiskos
NCSR Demokritos, Greece
Email: karaiskc@outlook.com

***Abstract*—The *Human-Robot Cloud* has previously been introduced as a framework for the creation of distributed, on-demand, reconfigurable human-machine cognitive systems[1]. These systems are made up of *sensing*, *processing*, and *actuation* components that are not limited to a specific type of application and potentially can be extended to multiple domains and may cover spatially smaller or larger areas. In this paper, we revisit the Human-Robot Cloud architecture and present its pilot deployment on the campus of NCSR Demokritos, a research institution in Greece. In particular, our concrete deployment aims to be demonstrated in three specific application scenarios; namely, *Human-Aware Smart Buildings with Energy Optimization*, *Security and Surveillance*, and *Smart Tour Guide System*. In this paper, we present in detail an example implementation of the Smart Buildings scenario: a real-world application with immediate benefits in energy optimization and energy savings. Environmentally sensitive issues, such as the ground-up development of energy efficient buildings or reducing the environmental impact of the existing infrastructure, has received much attention in the past. However, the traditionally offered solutions are central, non-transferable to other infrastructure, non-scalable and suffer from *single points of failure*. On the contrary, in this work, which is based on a specialization of the generic Human-Robot Cloud architecture, we attempt to move beyond the industrially available solutions to meet the requirements for scalable, reconfigurable and redistributable sensory, processing, and actuation units within buildings. A set of cameras, laser range finders, and other sensors, together with a number of processing and actuation elements, including face detection, expression recognition, and people trackers, are transformed to a prototypical reconfigurable distributed extended cognitive system, which can support multiple applications in the future.**

I. Introduction

Traditional artificial cognitive systems (for example, intelligent robots) share a number of common limitations. First, they are usually made up only of machine components; humans are only playing the role of user or supervisor. And yet, there are tasks in which the current state of the art of AI has much worse performance or is more expensive than humans: thus, it would be highly beneficial to have a systematic way of creating systems with both human and machine components, possibly with remote non-expert humans providing snippets of some seconds of their capacities in real-time. Second, their components are specific and dedicated to one and only one system, and are often underutilized for significant fractions of their lifetime. Third, there is no inherent support for robust, fault-tolerant operation, and if a new component becomes available, with better performance and/or cheaper cost, one cannot easily replace the old component. Fourth, and quite importantly in terms of their economics, they are viewed as a resource that needs to be developed and owned, not as a utility; i.e. not as a service provided on demand [2].

Motivated by the above state of affairs, the Human-Robot Cloud (CLIC) framework was presented in [1], [2]. CLIC is a framework for constructing cognitive systems that overcomes the above mentioned limitations. With the four-layer software architecture of CLIC, specific yet extensible mechanisms enable the creation and operation of distributed cognitive systems that fulfill the following desiderata: First, that are distributed yet situated, interacting with the physical world though sensing and actuation services, and that are also combining services provided by humans as well as services implemented by machines. Second, that are made up of components that are time-shared and re-usable across systems. Third, that provide increased robustness through self-repair mechanisms. Fourth, that are constructed and reconstructed on the fly, with components that dynamically enter and exit the system, while the system is in operation, on the basis of availability, pricing, and need. Quite importantly, fifth, the cognitive systems created and operated by CLIC do not need to be owned and can be provided on demand, as a utility – thus transforming human-machine situated intelligence to a service, and opening up numerous interesting research directions and application opportunities.

A similar framework is proposed by "Synaisthisi", a Greek-European funded project that aims to bridge the gap between the research area of reliable sensing technologies, optimal distribution, and network management with the information processing researchers abstracting from the sensing technologies and focusing on the data processing models, decision support and event recognition systems. "Synaisthisi" is an ongoing project with a strong focus in delivering applied, demonstrable frameworks and solutions for smart buildings applications. Towards the full implementation of "Synaisthisi", a pilot deployment of the basics of the framework is taking place in the campus of NCSR Demokritos. In this deployment, several sensing, processing, and actuation components are being created. The initial deployment aims to be able to be dynamically reconfigured in order to support three applications: Human-Aware Smart Buildings with Energy Optimization, Security and Surveillance, and Smart Tour Guide System. In this paper, we present an initial implementation of the Smart Buildings for Energy Optimization application.

978-4799-8422-0/15/$31.00

In this paper, we build on the need to improve the energy efficiency of buildings through leveraging the power of independent, distributed components that can be fused and be easily reconfigured in multiple applications to address different needs, e.g., security and surveillance applications or guided tours. Section II discusses the basic concepts behind the CLIC framework and a related application in the research area of smart buildings. In Section III, we discuss the requirements for the *messaging architecture* that will be responsible for the robust communication between the components. Section IV contributes to the technical specification of our system with details that would allow potential users to design and to deploy similar systems in the future. Section V is dedicated to the processing units, in particular the people counting application from video sequences, that were specifically developed for this application. Finally, Section VI concludes this work and discusses our future directions.

II. Overview and Basic Concepts

Mavridis *et al.*[2] introduced a framework that contributes towards improving the above-described state of affairs. In particular questions were asked such as: How can the cognitive systems of the future exhibit improved characteristics? Can they consist also of human components, given that for some cases they exhibit better performance and/or are more readily available than electronic elements? For example: humans are much better in performing activity recognition as compared to the state-of-the art automated systems (better performance). Another example is that of a human observer next to a broken traffic camera, who can be useful acting as a sensor that reports traffic conditions (availability). Based on this the following questions are raised up: Can the cognitive systems of the future include distributed components that are time-shared and re-used for various systems, and also enable much higher robustness and flexibility? For example why should the surveillance cameras of a city be dedicated only to surveillance, and why can they not be reused for other purposes too? Can we even reach a stage, where one is able to offer situated cognitive systems as an on-demand service, on the basis of the type of the requests? All of this seems like quite distant from the state of the art of today — but is it?

An interesting experiment trying to address these needs was the DARPA 2009 Network Challenge, often referred to as the Ten Red Balloons competition. During this Challenge, ten large red balloons were placed in locations around the United States, with their location unknown to the participating teams. The goal was to create a system that is able to locate the balloons in minimum time, without restricting the use of humans in the system. A team from the MIT Media Lab won the challenge: through an ingenious scheme thousands of non-expert humans were recruited lending some seconds of their eyes to the resulting system; this information was propagated and combined, in order for the system to determine the location of the balloons [3]. One can view the system that was created as a massive distributed cognitive system: with sensing (vision) provided by human components, pattern recognition (red balloon recognition) provided by humans too, and information fusion as well as propagation provided by electronic components. Notice that in this system the components are human as well as machine, they are distant (spread over a large geographical area). Furthermore, the human components are not dedicated to the system (i.e. the humans that spent 10 seconds of their time looking around for a balloon are also using their eyes and brains for other tasks), i.e. their sensing as well as cognition apparatus is time-shared and re-used. Finally, there is a large degree of robustness to the system as false-reports can be crossed-out of the system through the special algorithms used and through the inherent redundancy in sensing resources.

Another similar example is Von Ahn's [4] CAPTCHA-breaker scheme. CAPTCHAs are often used to prevent spam email programs and other bots from creating thousands of email accounts. They are usually strings of letters and numbers, with character sets that contain geometric distortions and occlusions. The characters in the CAPTCHAs are very easy for humans to recognize; however, they are quite difficult for machines, given the state of the art of Optical Character Recognition. A solution towards breaking CAPTCHAs, involves finding non-expert humans online, and incentivizing them, so that they break the CAPTCHA (by recognizing the characters) with the answer collected by the spam mailer program, which opens the accounts right away. The humans effectively lend some seconds to the system to perform the cognitive service of character recognition for it, incentivized by illegal downloads that the system offers in return for their services. In essence, a large distributed cognitive system is effectively created consisting of human as well as machine components, which dynamically enter and exit the system to achieve superior results that would have been impossible by either alone.

Finally, we consider a third recent development, the cloud computing paradigm. Traditionally, computation required ownership of resources: computers, storage space, and software. With cloud computing, computation is viewed as a "utility". In a similar sense with modern power and water networks, users of the cloud do not need to own the means of production or distribution (i.e. power generators, water sources and distribution networks): they just connect to the cloud, and time-share reusable distant distributed computation, storage, and code resources, in a transparent fashion (not knowing the whereabouts or the specifics of them), and with high robustness.

The aim of CLIC is to provide a conceptual framework to integrate ideas, constructs, architectures and techniques from human-machine cognitive systems, artificially intelligent agents and services of the kind presented in cloud computing to build human-machine cognitive and intelligent systems on demand. The contribution of the proposed framework is the identification and conceptual definition of four layers that need to be available for building cognitive systems applications.

A. Related Work

The area of Smart Building applications is quite wide, however, there has been little research on the redistributable and reconfigurable aspects of our proposed system. The most closely related work in the field is the *Building Operating System Services* (BOSS) [5] but we are not aware of other research that is trying to address the same issues within a complete framework that includes generic sensing, processing and actuation units.

Dawson *et al.* [5] recently presented BOSS, i.e., a set of operating system services that offer the ground on which users are able to quickly develop applications for managing large commercial buildings. BOSS proposes and implements an architecture for development of robust and portable applications. Similar to our proposed work, the applications are portable in the sense that the user has access to a set of services and APIs without interfering with the low level hardware issues of sensors and actuators. However, the most interesting contribution of this work is the introduction of an approximate query language that specifies the relationship of components and not specific devices. Furthermore, they introduce a reasoning system about failure cases and present a time series service that treats historical and real-time data to be treated uniformly. The latter functionally is freely available from ROS in our proposed work and the user is able to log and replay the logs as if they were real-time. Furthermore, our work is extensible towards full reconfigurability.

As attractive as the BOSS framework may be, there is a lack of the human element in either the sensing, processing or actuation units. BOSS briefly mentions that the human feedback could be used in the system. However, in our work, we have a clear understanding of the human element either as a sensing, processing, or actuation unit as this system has derived from the Human-Robot Cloud. Our application may have not reached the same level of maturity with BOSS, however, our testbed is ready for the deployment of a more complex system that will include human and robotic elements within its units.

III. Requirements for the MOM

A major constraint for such applications is either the use of an established system or the bespoke development of a *Message Oriented Middleware* (MOM). The MOM will support redistributable and reconfigurable sensing, processing and actuation units through a generic middleware that will be responsible for the intercommunication of these systems, in order to achieve the required level of control for any kind of buildings and applications. "Synaisthisi" project [6] has defined the minimum requirements for any system that would support the easy deployment of our system in buildings of any type, either commercial or private housing, and of any scale. These requirements are iterated below.

1) The independent components are distributed and unaware of the internal specifics of their lateral sensing, processing and actuation network. Hence, the MOM must support message passing amongst the components and, more specifically support messages with custom types to facilitate any type of data that may be the output of the sensors. For example, float values, videos from cameras or point clouds should all be supported by the system.
2) The nodes of the system must be easily connected within a network by assigning them dynamic IP addresses.
3) User needs may include blocking as well as non-blocking API calls to each component of the system.
4) Support for numerous message passing channels on each IP address
5) Ease of implementation of buffering schemes
6) Data throughput, delay times, and robustness which are adequate both for low-data rate non-time-critical messaging (such as temperature measurements), as well as high-data rate (such as video) and highly-time-critical measurements (for example, laser ranging data tied to a real-time motor-control loop
7) Support for components written in various programming languages (C++, Java, Python and more) and running in various operating systems (Windows and Linux, also support for iDevices or PDAs useful)
8) Availability of a large base of existing components supporting sensing, actuation, and processing services, including cameras, RGB-D sensors such as the Kinect, Laser Range Finders, but also various kinds of motors and output devices
9) Existence of message-passing wrappers for important application-oriented processing libraries such as OpenCV and Point-Cloud-Library
10) Ease of implementation of time-stamps and logging functions

Multiple MOM systems have been proposed in the literature as well as other robotics-oriented middlewares [7]. This work is specifically interested in the smooth integration of current and future robotic systems to actively participate in the architecture. Within the area of robotics a well established system, ROS: the Robot Operating System, satisfies the desiderata and furthermore allows the straightforward integration of robotic components for future applications; e.g., robotic tour guides in museums. ROS in its official page is promoted as "a set of software libraries and tools that help you build robot applications", however, ROS offers a lot more. ROS is ideal for distributed computing systems as its subsystems are converted to nodes that belong to a virtual computation graph. Nodes are highly reconfigurable to meet the needs of the user or the developer and extensive tools are available to satisfy most needs. ROS offers a publish-subscribe messaging framework that is generic and allows various types of custom messages to be communicated within its network. Furthermore, research groups around the world publicly share their ROS nodes implementing state-of-the-art algorithms that would serve as processing units in our system, e.g., computer vision algorithms. Last but not least, ROS offers an active ecosystem with strong interaction between its users through social media, on-line communities, fora and *ROSCon*, the annual ROS Developer Conference.

IV. Smart Buildings: A Case Study

Investing in energy efficiency is a compelling act for private and public institutions that will face increasing energy demands in the near future. In particular, *Architecture 2030*, a non-profit organization, reports that the US buildings in general are responsible for approximately 75% of the total US energy consumption every year. Considering that the average consumption difference in green buildings is about 30% less than the conventional buildings, it highlights the heavy impact of any improvements that would have to the world energy consumption [8]. A different manifestation of transforming a building to *green* is improving its energy efficiency through the installation of sensing, processing, and actuation units in the building. Recently, Dawson *et al.* [5] emphasized the importance of reusing existing components of traditionally built buildings, e.g., ventilation systems, fire alarms, heating and cooling systems, in an interoperable way to extend the existing infrastructure.

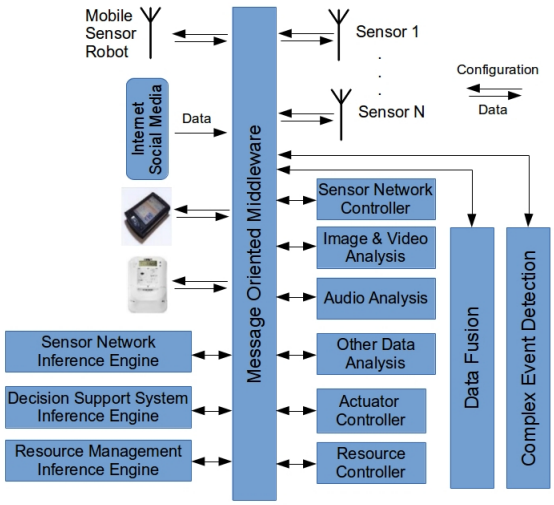

Fig. 1. Typical classes of Sensing, Processing, and Actuation components that can be reconfigured to support all three application scenarios or other complex domains.

The National Centre for Scientific Research, NCSR Demokritos, is the biggest research centre in Greece. It currently employs more than 1000 researchers, engineers, technicians and administrative personnel. The number of people along with the various activities, e.g., frequent meetings and workshops, that take place within the research centre create the perfect environment for a case study of our system. It is also a great testbed since the occupants are familiar with the latest technologies and would welcome such a system in their daily life. Finally, we expect the deployment of our system will have immediate return in terms of savings given the scale of the campus. For our pilot experiment we have used a single room to demonstrate its capabilities before scaling it up to the building and eventually the campus. Below we present the sensing, processing and actuation units that were used, while the general framework can be seen in Figure 1.

A. Sensing Units

1) Hardware: In the proposed testing environment, all sensing units are attached to a Raspberry Pi (Pi) computer. The low cost Pi converts any sensor to a network sensor enabling it to perform basic information processing as well as run ROS natively. ROS on the Raspberry Pi is not currently officially supported, however, it is possible to port it and deploy it with ease as an image to other Pis. Essentially, the Pi becomes one or more portable ROS nodes with the ability to process sensory input from different types of sensors connected to it. In the current implementation a single Pi is responsible for all the processing of our sensors, i.e., 6 sensors providing a total of 11 independent measurements. The current setup provides redundancy by having two temperature/humidity and two luminosity sensors connected to the same Pi. In future implementations, these sensors will be connected to multiple Pis (ROS nodes) and will be deployed on different places within the same room, or different rooms within a building. The sensor list and measurements provided include an RHT03 (or DHT22) Temperature/Humidity sensor, an AM2315 Temperature/Humidity sensor, two TSL2561 Luminosity sensors (Broadband, Lux, IR), and two *Clever* BWBL-2B05M Door Passage Detectors.

The RHT03 sensor offers 0-100% humidity readings with 2-5% accuracy and -40 to $125°C$ temperature readings with $\pm 0.5°C$ accuracy. The sampling rate is limited to no more than 0.5Hz. The AM2315 sensor offers 0-100% humidity readings with 2% accuracy and -40 to $125°C$ temperature readings with $\pm 0.1°C$ accuracy. The TSL2561 sensor operates at 400kHz (I2C Fast-mode) and automatically rejects 50/60Hz Lighting Ripple. The BWBL-2B05M door passage detector is composed of an IR transmitter and receiver that fires an alarm when interrupted, i.e., activates a relay switch (40ms trigger response time) that is monitored by the Pi. Two sets of sensors are carefully positioned in parallel to each other at a distance of about 40cm, to eliminate the possibility of cross-illumination. Two sensors are deployed in order to register whether a person is walking in or out of a room based on the firing time difference. Hence, it is possible to update the count of people at any moment in the room, however, not accurately due to physical limitations of the relay switches or multiple people passing simultaneously. All sensors are not plug-and-play and custom circuitry was designed to interface them with the Pi.

2) Software: A required step for each sensor is to create a ROS wrapper/driver to make the sensory readings available across the ROS computational graph. Each sensor is individually interfaced with ROS and is published as a ROS topic. ROS wrappers were created over the python2 drivers of the sensors, and also a UDP connection was used in one case due to a driver incompatibility. Other sensors have also been used in our application such as a webcam, a Kinect sensor, and an LMS100 laser range finder, which are directly supported by ROS Hydro. For the Kinect, the "openni" package was used.

3) ROS Network: In order for such a system to be redistributable and reconfigurable the sensors have to be arranged as multiple ROS nodes, either on Pis, desktop PCs, laptops, or other ROS-equipped systems. In the pilot setup, a master node is running *roscore*, which acts as a DNS server for all other nodes. The remaining machines in the network have access to a list of published ROS *topics*. Any user or agent in the network may subscribe to any number of ROS topics to further use the sensory information. An example of a central application fusing all of the incoming sensor data is presented later. A shortcoming of the current implementation is that any failure in the master node will result in unexpected behavior in the system. Hence, relying on a single machine running "roscore" introduces a single-point of failure. In the future, we plan to resolve this issue with an extension where roscore will be watchdogged and redundant agents will be initiated on demand on different machines that will take over the role of "roscore".

4) Application: A demo application is demonstrated in Figure 2. A requirement of the application is to remain ROS independent, therefore, additional ROS de-wrappers were developed. ROS de-wrappers are also ROS nodes that convert incoming ROS messages into traditional data types. Hence, ROS is only responsible for the data communication. The application reads the sensor data, processes it according to previously setup rules following international standards, and fires actuator responses to the processed signals. Unfortunately, due to the lack of actuation elements we were not able to

demonstrate this aspect, however, we present the pilot by printing the corresponding actions as messages on display.

V. Processing Units: Counting People from Video

We propose a fast real-time method for automatic detection and tracking of multiple people in an indoor environment (closed room), from video captured by a monocular fixed uncalibrated camera.

Visual detection and tracking of people is a well-known computer vision problem that is notoriously challenging due to the variations in human appearance, body poses and viewing angles. However, because real-time performance is important to our system, and because we only need to solve this problem for the constrained indoor environment of a small closed room (hence small number of people with restricted movement), we have opted for a compromise between algorithmic simplicity and accuracy (false detection rate) in the design of our method [9], [10], [11], [12].

Our method combines three sources of information in a single image: motion, appearance (color), and the results of state-of-the-art fast person detectors. Specifically, multiple person detectors (of the full body, face, head & shoulders, and lower body) are applied independently in each frame, and their results are used to update the state of the currently tracked targets. In order to handle cases of missing person detections, we also rely on motion information (in the form of blobs detected via adaptive background modelling and subtraction) and color histogram-based mean shift tracking [13]. The method hence consists of three main processing modules, namely 1) Person detection 2) Motion detection (or change detection) 3) Object tracking. The implementation was done in C++ using the open-source OpenCV library and currently runs at approximately 10fps.

A. Person Detection Module

The goal of the Person Detection Module is to detect as many people as possible in a given single image, e.g., see Figure 3, independently of other images in the video. Five different detectors are applied in parallel for this task : 1) Frontal face detector, 2) Profile face detector, 3) Head and shoulders detector, 4) Lower-body detector, and 5) Full-body detector. The first four detectors are based on the method by Viola and Jones [14], which uses Haar-like filters (wavelets) and multiple cascades of boosted classifiers. The full-body detector is based on the method by Dalal and Triggs [15], which uses a Histogram of Oriented Gradient (HOG) descriptor and a linear Support Vector Machine (SVM) classifier.

Obviously this parallel detectors approach is more robust than using a single detector, since people are often sitting or occluded (by other people or objects in the scene), as well as because these detectors are not perfectly accurate; in fact they typically suffer from high false alarm rate.

We subsequently merge the results of these different detectors by clustering of the bounding boxes (rectangles) into disjoint groups with similar sizes and similar locations. This is a crucial step in our processing pipeline as it greatly reduces the complexity of the subsequent tracking step. The output of this module hence consists of a set of person detections.

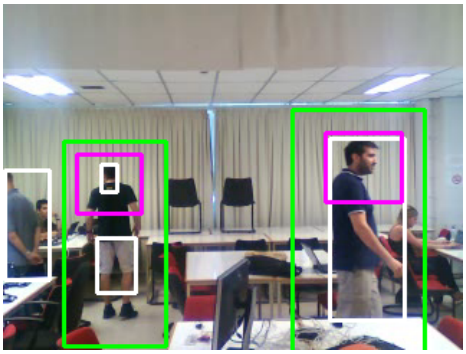

Fig. 3. Single frame of the people counting processing unit in the main room that our system was deployed.

B. Motion Detection

The goal here is to detect motion regions in an image frame. Since a static camera is assumed, we use an adaptive mixture-of-Gaussians (MOG) background model for this task. A set of motion regions (or blobs) is obtained in each frame by first subtracting the frame from the background model, then applying some elementary morphological operations (erosion and dilation) in order to reduce noise in the binary image, and finally doing a connected components analysis to separate the different motion regions.

C. Object Tracking

Given the set of person detections and blob regions from the previous two modules, the goal here is to integrate this information to update the state of the current set of tracked targets (people in the scene). Obviously, this is a classic data association (correspondence) problem, wherein we seek a best match between new detections and tracked targets. A heuristic algorithm is used that consists of the following steps :

1) Predict the new image location of each tracked target based on a simple constant velocity motion model. The velocity is estimated every 20 frames using a batch least squares fitting.
2) Determine best matching between new detected persons and current tracked targets based on simple bounding box proximity. This step results in 4 cases: 1-1 matches, unmatched targets, unmatched detections, many-many matches.
3) Handle each unmatched target by trying to match it with a motion blob, and if that fails, then use *mean shift* tracking based on a simple appearance model of the person's upper body part. The latter model is obtained by extracting a 2-dimensional colour histogram using the a* and b* components of the LAB color space.
4) Handle each unmatched detected person by creating a new tracked target for it.

Note that we currently do not handle many-many matches since they are ambiguous cases that require additional cues to be resolved effectively.

VI. Conclusion

We have introduced the pilot deployment of a Smart Buildings application in the campus of NCSR Demokritos. Initially,

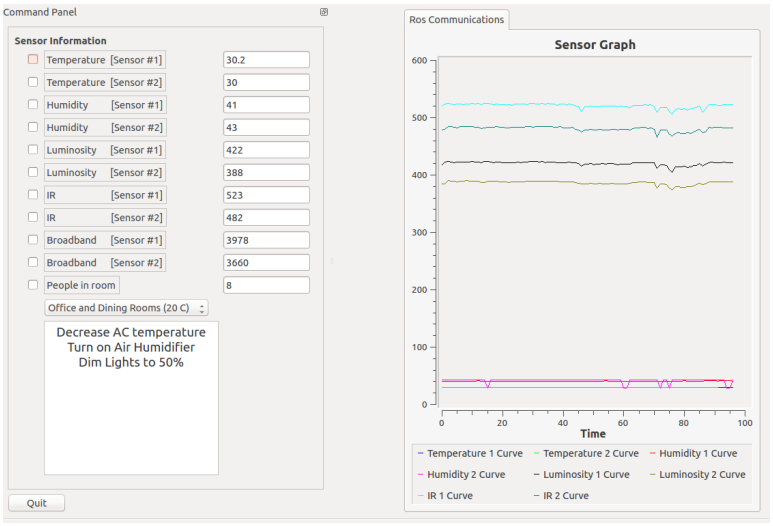

Fig. 2. The demo application subscribes to the available ROS topics within its network and provides an intuitive interface. The system is able to follow simple rule-based code to force actuation suggestions that a human may follow, or in the future autonomously take actions that will save energy or offer a more comfortable working and living environment.

we briefly discussed the CLIC framework and "Synaisthisi" project and discussed the requirements for a Message Oriented Middleware that would allow their utilization in a realistic environment. In the current deployment, several sensing, processing, and implied actuation components were created focusing on the energy saving aspect of smart buildings. However, it has been discussed how it is possible to reconfigure the nodes into a different future application, such as Security and Surveillance, or a Smart Tour Guide System. This work is focused mostly on the lower level architecture of the system and the actual implementation offering valuable information to potential users that want to deploy ROS as part of their Smart Buildings applications. In the future, we plan to offer a higher level application that will offer a more challenging testbed, however, this paper is an important step towards that direction as it covers important aspects of the actual implementation. In particular, we plan to extend the sensing capacity with microphone arrays and the integration of all the sensing units in a complex application that fuses them to facilitate higher levels of inference.

Acknowledgment

The authors would like to thank Dr. Constantinos Spyropoulos for his helpful comments and support. The research leading to these results has received partial funding from the Greek General Secretariat for Research and Technology and from the European Regional Development Fund of the European Commission under the Operational Program "Competitiveness and Entrepreneurship" (OPCE II) - Priority Axis 1 "Promotion of innovation, supported by research and technological development" and under the Regional Operational Program Attica - Priority Axis 3 "Improving competitiveness, innovation and digital convergence".

References

[1] N. Mavridis, T. Bourlai, and D. Ognibene, "The human-robot cloud: Situated collective intelligence on demand," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*. IEEE, 2012, pp. 360–365.

[2] N. Mavridis, S. Konstantopoulos, I. Vetsikas, I. Heldal, P. Karampiperis, G. Mathiason, S. Thrill, K. Stathis, and V. Karkaletsis, "Clic: A framework for distributed, on-demand, human-machine cognitive systems," in *arXiv preprint arXiv:1312.2242, currently submitted at Journal of Cognitive Systems Research*, 2013.

[3] J. C. Tang, M. Cebrian, N. A. Giacobe, H.-W. Kim, T. Kim, and D. B. Wickert, "Reflecting on the darpa red balloon challenge," *Communications of the ACM*, vol. 54, no. 4, pp. 78–85, 2011.

[4] L. Von Ahn, B. Maurer, C. McMillen, D. Abraham, and M. Blum, "recaptcha: Human-based character recognition via web security measures," *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[5] S. Dawson-Haggerty, A. Krioukov, J. Taneja, S. Karandikar, G. Fierro, N. Kitaev, and D. E. Culler, "Boss: Building operating system services." in *NSDI*, vol. 13, 2013, pp. 443–458.

[6] "D2.3, "System architecture" Project - SYNAISTHISI, Action - KRIPIS, OPCE II, Ministry of Development and Competition, 2014," NCSR Demokritos, Athens, Greece, Tech. Rep., 2014.

[7] N. Mohamed, J. Al-Jaroodi, and I. Jawhar, "A review of middleware for networked robots," *International Journal of Computer Science and Network Security*, vol. 9, no. 5, pp. 139–148, 2009.

[8] G. Kats, L. Alevantis, A. Berman, E. Mills, and J. Perlman, "The costs and financial benefits of green buildings," *A Report to Californias*, 2003.

[9] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008.

[10] B. Schiele, M. Andriluka, N. Majer, S. Roth, and C. Wojek, "Visual people detection: Different models, comparison and discussion," in *Proceedings of the IEEE ICRA Workshop on People Detection and Tracking*, May 2009.

[11] Z. Jiang, D. Q. Huynh, W. Moran, S. Challa, and N. Spadaccini, "Multiple pedestrian tracking using colour and motion models," *Digital Image Computing: Techniques and Applications*, pp. 328–334, 2010.

[12] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. Van Gool, "Online multiperson tracking-by-detection from a single, uncalibrated camera," *IEEE Trans. Pattern Analysis Machine Intelligence (PAMI)*, vol. 33, no. 9, pp. 1820–1833, 2011.

[13] D. Comaniciu, V. Ramesh, and P. Meer, "Real-time tracking of non-rigid objects using mean shift," 2000.

[14] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, 2001.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886–893.