Loughborough
University

This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/

A blueprint for integrated eye-controlled environments

D. Bonino[1], E. Castellina[1], F. Corno[1], A. Gale[2], A. Garbo[1],

K. Purdy[2] F. Shi[2],

[1] Politecnico di Torino, Dipartimento di Automatica e Informatica

[2] Loughborough University, Ergonomics and Safety Research Institute, Applied

Vision Research Center

Abstract   Eye-based environmental control requires innovative solutions for supporting effective user interaction, for allowing home automation and control, and for making homes more "attentive" to user needs. Several approaches have already been proposed, which can be seen as isolated attempts to address only partial issues and specific sub-sets of the general problem.

This paper aims at tackling gaze-based home automation as a whole, exploiting state of the art technologies and trying to integrate interaction modalities than are currently supported and that may supported in a near future. User-home interaction is sought through two, complementary, interaction patterns: direct interaction and mediated interaction. Integration

between home appliances/devices and user interfaces is granted by a central point of abstraction/harmonization called House Manager.

Innovative points can be identified in the wide flexibility of the approach which allows on one side to integrate virtually all home devices having a communication interface, and, on the other side, mixes-up direct and mediated user interaction exploiting the advantages of both. A complete discussion about interaction and accessibility issues is provided, justifying the presented approach from the point of view of human-environment interaction.

1 Introduction

The challenge of intuitive and comprehensive eye-based environmental control system requires innovative solutions on different fields: user interaction, domotic system control, image processing. The current available solutions can be seen as "isolated" attempts at tackling partial sub-sets of the problem space, and provide interesting solutions in each sub-domain.

This paper seeks to devise a new-generation system, able to exploit state-of-the-art technologies in each of the fields and anticipating interaction modalities that might be supported by future technical solutions in a single integrated environment. In particular, the paper presents a comprehensive solution, in which integration is sought along two main axes: (a) integrating various domotic systems and (b) integrating various interaction methodologies.

The intelligent devices adopted in current intelligent environments, and those that can be foreseen in future high-tech homes, are characterized by a high variability in terms of features, connectivity, funtionality, etc. The lack of de-facto standards, despite the existence of several industrial consortia, generated a proliferation of different domotic systems (EIB/KNX [1], BTicino MyOpen [2], X10 [3], LonWorks [4], ...) able to connect different families of devices. Besides domotic systems, we are also witnessing the proliferation of other kinds of intelligent devices, that are not part of specific infrastructures, but are stand-alone devices, usually equipped with some form of network connectivity (Wi-Fi, Bluetooth, Ethernet, Infrared, . . .). These standalone devices range from surveillance sensors or cameras, to PC-like media or entertainment centers. The comprehensive solution we are seeking should be able to manage this Pandora's box of device characteristics, features, networks, and open and proprietary protocols.

On the other hand, interaction methodologies should take into account the latest results in human-environment interaction, as opposed to human-computer interaction. The paradigm of "direct interaction", so familiar in desktop environments and now also extended on the Internet with Web 2.0 applications, is not so natural when applied to environmental control. Selecting a user interface element that represents a physical object, that is also within the user's view field, is quite an indirect interaction method. Directly "selecting" objects by staring at them would be extremely more direct and intuitive. Besides the technical difficulties of detecting the object(s) gazed

by the user, there is a design trade-off between the more direct selection and the traditional mediated interaction. While direct interaction eases object identification but leaves few options for specifying the desired action, mediated selection, where the object is selected on a computer screen, complicates object selection but allows an easy selection of the desired commands. In addition, mediated selection allows to interact with objects that are not directly perceivable by the user like thermal control, automated activation of home appliances or objects in other rooms. The comprehensive solution proposed in this paper seeks the appropriate trade-off among these opposite interaction methods, proposing a system able to support both, and to integrate them thanks to the aid of portable devices.

The overall vision is centered on a "house manager" that, on one side, builds an abstract and operable model of the environment (described in section 4) by speaking with different domotic systems according to their native protocol, and with any additional existing device. On the other side, it offers the necessary APIs to develop any kind of user interface and user interaction paradigms. In particular in this paper we will explore eye-based interaction, and will compare "mediated" menu-driven interaction (section 5.1) with innovative "direct" interaction (section 5.2).

The paper is organized as follows: in section 2 some relevant related works are discussed, reporting state of art solutions for gaze-based home interaction. Section 3 introduces the general architecture of the proposed approach. Section 4 describes in details how different home devices and

domotic networks can be integrated and made interoperable through the House Manager component. Section 5 compares the two, gaze-based, interaction modalities, highlighting the pros and cons of both and analyzes how the two can be successfully integrated. Eventually section 6 provides conclusions and proposes some future works.

2 Related works

Vision is a primary sense for human beings; through gaze people can sense the environment in which they live, and can interact with objects and other living entities [5]. The ability to see is so important that even inanimate things can exploit this sense for improving their utility. Intelligent environments, for example, can exploit artificial vision techniques for tracking the user gaze and for understanding if someone is staring at them. In this case they become "attentive" being able to detect the user's desired interaction through vision [6].

Several eye-gaze tracking techniques are described in literature. The most prevalent are pupil tracking, electro-oculography, corneal and retina reflection, artificial neural networks, active models and other methods. A general summary of the most adopted methods can be found in [7,8].

These techniques are variably used in several commercial systems that provide assistive input interfaces for disabled people. A complete and updated list of such tools is provided in [9]. Thanks to the COGAIN network, some researchers and some producers of commercial trackers are currently

working together to define a new, universal, and standard API for eye control applications [9], enhancing the interoperability of gaze-based assistive technologies.

Gaze tracking technologies are usually adopted for providing alternative user interfaces for PC applications, in particular typesetting or Augmentative Alternative Communication (AAC) applications. However this kind of interaction can also be used in other contexts, such as home automation.

Home automation is a quite old discipline that today is gaining a new momentum thanks to the ever increasing diffusion of electronic devices and network technologies. Currently, many research groups are involved in the development of new architectures, protocols, appliances and devices [10]. Also commercial solutions are increasing their presence on the market and many brands are proposing very sophisticated domotic systems like the BTicino MyHome [11], the EIB/KNX [1], which is the result of a joint effort of more than twenty international partners, the X10 [3] and the LonWorks [4] systems.

Recently, literature reports some research about eye-gaze-controlled *intelligent* environments. In these studies two main iteraction modalities are foreseen: direct interaction and mediated interaction. In direct interaction paradigms gaze is used to select and control devices and appliances either with head-mounted devices that can recognize objects [12] or through "intelligent" devices that can detect when people stare at them [6]. Using mediated interaction, instead, people control a software application (hosted

on desktop or portable PCs) through gaze, thus being able to control all home appliances and devices [13].

While being interesting and sometimes very effective, the currently available solutions only try to solve specific sub-problems of human-environment interaction, focusing on single interaction patterns, interfacing a single or few home automation technologies. This paper, instead, aims at integrating different interaction patterns, possibly exploiting the advantages of all, and aspires to interoperate with virtually every domotic network and appliances. The final goal is to provide a complete environment where the user can interact with his house using the most efficient interaction pattern depending on his abilities and on the kind of activities he wants to perform.

3 General architecture

Mixing interaction by gaze and home automation, requires an open and extensible logic architecture for easily supporting different interaction modalities, on one side, and different domotic systems and devices on the other. Several aspects shall be in some way mediated, including different communication protocols, different communication means, different interface objects. Mediation implies, in a sense, centralization, i.e., defining a logic hub in which specific, low level aspects are unified and harmonized into a common high-level specification.

In the proposed approach, the unification point is materialized by the concept of a "house manager" which is the heart of the whole logic archi-
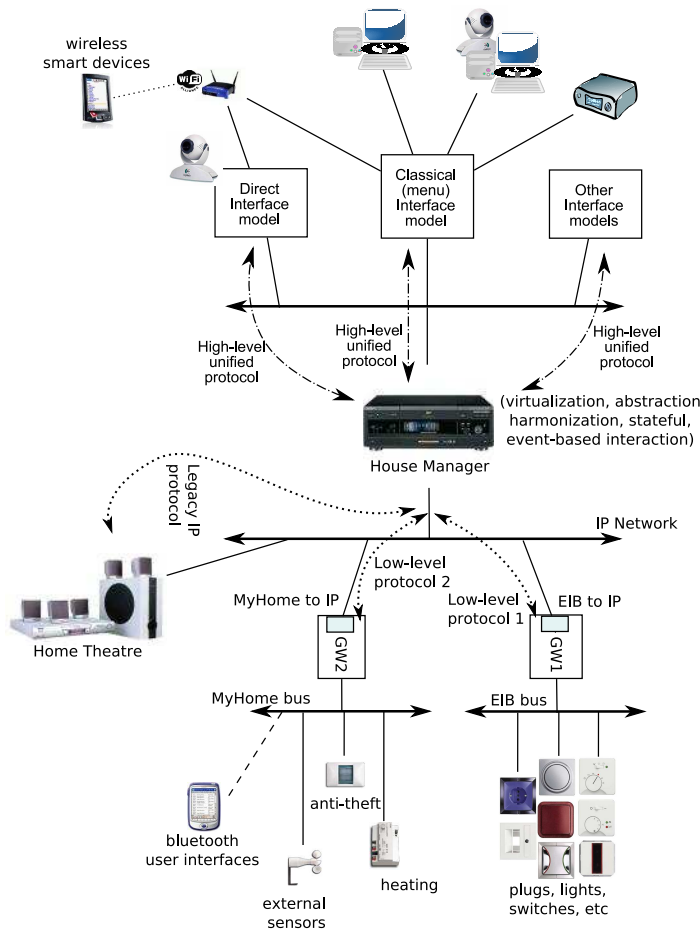
Fig. 1 The general architecture of the proposed system

tecture (Figure 1) and acts as gateway between the user and home environments.

On the "home side" the manager interfaces both domotic systems and isolated devices, capable of communicating over some network, through the proper low level protocols (different for each system). Every message on this side is abstracted according to a high-level semantic representation of the home environment and of the functions provided by each device. The

state of home devices is tracked and the local interactions are converted to a common event-based paradigm. As a result, low level, local events and commands are translated into high-level, unified messages which can be exchanged according to a common protocol.

On the application side, the high level protocol provided by the manager gives home access to several interface models, either based on direct or mediated interaction. Two main models are discussed in this paper, the first based on attentive devices and the second based on a more classical menu-based interface.

4 Integrating domotic systems

In order to provide a suitable way for interfacing user interfaces with domotic networks, a common access point shall be designed, able to seamlessly interact with different domotic standards and devices. The main features required to such an access point are:

1. the ability to interface virtually every domotic network;

2. the ability to provide access to domotic devices through a simple, high-level, unified protocol;

3. the ability to interface any kind of device that can be remotely controlled (Hi-Fi systems, DVD players, media-centers);

4. the ability to enable cross communication between different domotic devices and networks;

5. the ability to provide access through well-defined, standard APIs (Web
   Services as an example);

In the proposed approach, these features are implemented by a module
called House Manager [14] that becomes the central point for interaction be-
tween user interfaces and the home (Figure 1). The House Manager's main
task is to abstract specific domotic protocols to a high-level, uniform rep-
resentation, that integrates in a common format all the information about
the house (control procedures, appliances, furniture, layout,...). Such a uni-
form representation can be easily obtained through the DomoML [15] set
of ontologies and communication languages, specifically designed for house
environment modelling.

DomoML provides on one side a complete, formal and flexible repre-
sentation scheme for home environments and on the other side it defines a
XML-based high-level communication language, independent from specific
domotic infrastructures. Representation is formal since it is based on widely
adopted Semantic Web (SW) standards such as OWL and RDF/(S) that
can be mapped to first-order logic statements. This allows both leverage of
mature technologies from the SW and integration advanced reasoning facil-
ities that can help in building the home "intelligence". DomoML models a
home environment both positionally and functionally; three main ontologies
compose the DomoML set named, respectively, DomoML-env, DomoML-fun
and DomoML-core (see Figure 2). DomoML-env provides primitives for the
description of all "fixed" elements inside the house such as walls, furniture
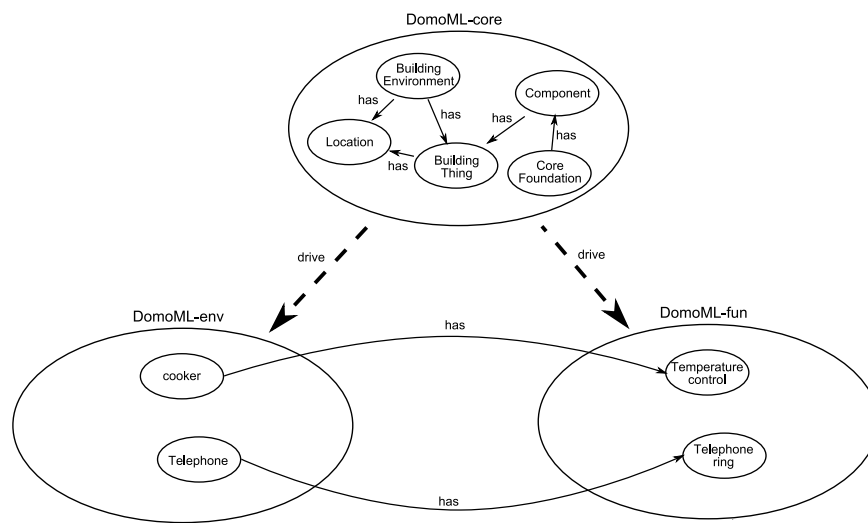
Fig. 2 The DomoML set of ontologies.

elements, doors, etc. and also supports the definition of the house layout by means of neighbourhood and composition relations. DomoML-fun provides means for describing the functionalities of each house device, in a technology independent manner. It defines basic controls such as linear, rotative knobs as well as very complex functions such as heating control and scenarios definition. DomoML-core, eventually, provide support for the correlation of elements described by DomoML-env and DomoML-fun constructs, including the definition of the proper physical quantities.

The internal structure of the House Manager is depicted in Figure 3 and is deployed as an OSGi [16] platform. OSGi implements a complete and dynamic component model where applications or components (coming in the form of bundles for deployment) can be remotely installed, started, stopped, updated and un-installed. This framework is becoming the reference model
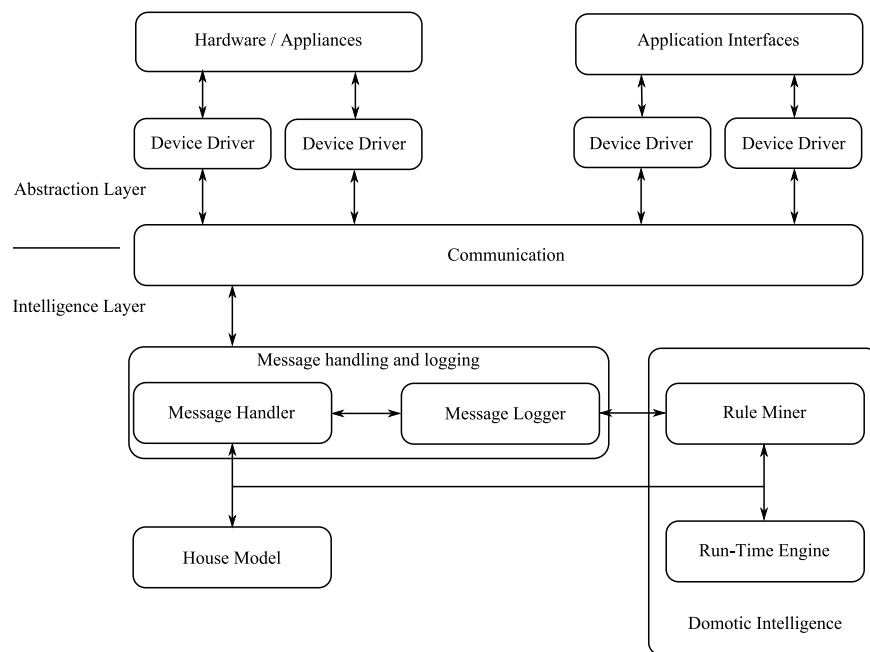
Fig. 3 The internal House Manager architecture.

for the integration of domotic networks as, in the domotic community vision, manufacturers will likely provide OSGi bundles for accessing each specific domotic infrastructure thus enabling easy interoperability. The House Manager architecture is roughly organized in two main layers: an abstraction layer and an intelligence layer.

The abstraction layer, which includes the device drivers, interfaces the controlled devices/environments and provides means for translating low level bus protocols into DomoML-com messages (Figure 4). Each domotic network, based on a different communication protocol, is managed by its own driver. A driver is implemented as an OSGi bundle, and must know how to translate low-level messages, understood by the network to which is con-

```
<Condition>

    <Name>PhoneCondition</Name>

    <ConditionAND>

        <FromDevice >SiemensT330</FromDevice>

        <Function>PhoneRing</Function>

        <FunctionStatus>on</FunctionStatus>

    </ConditionAND>

    <Action>

        <ToDevice>ElectricalCookerBauknect_ELZD5960</ToDevice>

        <Function>SwitchOff</Function>

        <FunctionStatus>off</FunctionStatus>

    </Action>

</Condition>
```

Fig. 4 A typical DomoML-com message.

nected, in DomoML-com constructs, and viceversa. Drivers can be loaded at runtime thus making the architecture flexible and extensible enough to manage many different domotic technologies.

Standalone devices having a communication interface can interact with the house manager by means of proper drivers, without requiring any changes in the manager architecture. As can easily be noticed, also application interfaces for hosting human interaction are seen as devices that can be connected to the manager by means of proper drivers.

The intelligence layer is organized in three interacting entities: the house model, the message handling and logging sub-system, and the domotic intelligence component. The house model represents every controllable, or sense-able device and supports the description of other house elements such as the walls, rooms and furniture. All the "fixed" elements take part in the house model definition by direct instantiation of prototypes defined by the DomoML-core and by the DomoML-env ontologies. Controllable, or sense-able, objects are, instead, modelled by instantiating prototypes defined by the DomoML-core, the DomoML-env and the DomoML-fun ontologies.

The message handling and logging subsystem has a two-folded nature, reflected by the functional blocks of which it is composed. The logging block persistently traces all the events and commands that occur in the house manager working time, providing support for diagnostic and for machine-learning algorithms that can leverage historical series of user behaviors and commands to partially automate or facilitate frequent actions. The message handling block, instead, acts as a router between the entities located at the abstraction layer. In particular, the message handling block listens for messages coming from drivers and, on the basis of the house model, decides to which other drivers such messages shall be routed (see Figure 5). Messages can be simply forwarded (routing) or can trigger further elaboration by the home intelligence component (ruled forwarding) that, in turn, can generate new messages to be handled. Besides being routed or elaborated, every
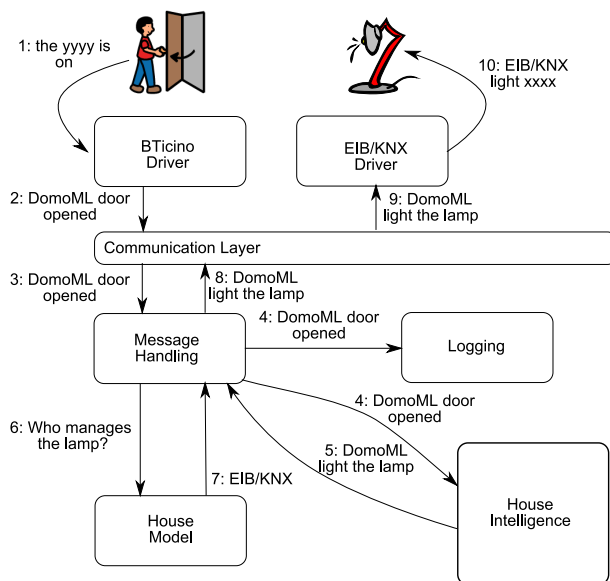
Fig. 5 The message handling interaction diagram.

message is also dispatched to the logging block for persistent tracing of commands and actions.

The domotic intelligence is mainly composed of two parts: the Rule Miner, which runs off-line learning of frequent actions from the manager logs and the Rule Engine, which operates at run-time by listening home and application events, and by taking the proper actions.

5 Human-Interaction paradigms

Users normally interact with the surrounding environment by manipulating physical objects, e.g., pulling up a lever for switching on the light, pushing a button for activating the dishwasher and so on. Interaction by object manipulation is sometimes infeasible, especially for users with physical im-

pairments or for elderly individuals. In such cases alternative methods of interfacing home appliances shall be provided.

Depending on the device to control, direct interaction, through gaze, or mediated interaction through menu-based PC applications may be preferrable. Devices with few operation modalities can be easily controlled by gazing at them, e.g., lights or doors, whereas more complex appliances may be better controlled using a sequence of menus on a PC screen. In both cases the main challenge is to define a clear and portable interaction pattern, common to both direct and mediated interfaces. In this way, the same tasks can be performed by either looking at the physical objects or at their proxies on a computer screen. The more natural is the solution provided the more effetive is the interface, limiting users' stress.

5.1 Mediated Interaction

Configuring, activating or simply monitoring complex appliances as well as complex scenarios can become really difficult by only gazing at them. In these cases a mediated interaction which allows to control the several aspects involved in these operations through a menu-based PC application can be more effective.

In the mediated interaction paradigm, gaze-based actions and reactions are accomplished through a menu-driven control application that allows users to fully interact with the domotic environment.
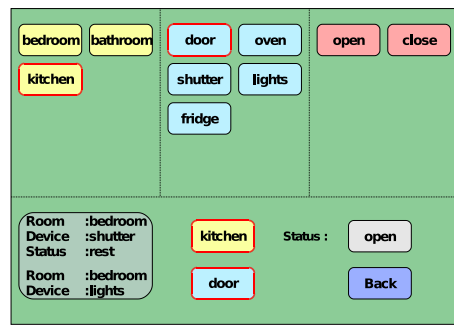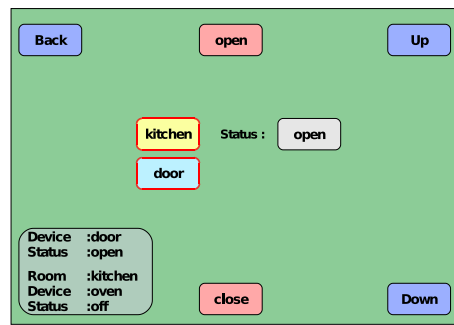
Fig. 6 The control application with a quite accurate tracker.

Such application shall respect some constraints, with respect to the different categories of users being expected. When users need a different application layout, related for example to the evolution of their imparment,they shall not be compelled to learn a different way of interacting with the application. In other words, the way in which commands are issued shall persist even if the layout, the calibration phase or the tracking mode changes.

To reach this goal the interaction pattern that drives the command composition has to be very natural and shall be aware of the context of the application deployment. For example, in the real world, if a user wants to switch on the kitchen light, s/he goes in that room, then s/he searches the proper switch and finally confirms the desired state change actually switching on the light. This behaviour has to be preserved in the control application command composition and the three involved steps must remain unvaried even if the application layout changes according to the eye tracker accuracy.

In this paper, mediated interaction can either be driven by infrared eye trackers (maximum accuracy/resolution) or by visible light trackers (web-

Fig. 7 The control application with a low-cost visible light tracker.

cam or videoconference cameras, minimum accuracy/resolution). These two extremes clearly require different visual layouts for the control application, due to differences in tracking resolution and movement granularity.

In the infrared tracking mode, the system is able to drive the computer mouse directly, thus allowing the user to select graphical elements as large as normal system icons (32x32 pixels wide). On the other hand, in the visible light tracking mode few areas (6 as an example) on the screen can be selected (on a 1024x768 screen size this would mean that the selectable area is approximately 341x384 pixels). As a consequence, the visual layout cannot remain the same in the two modes, but the interaction pattern shall persist in order to avoid the user to re-learn the command composition process, which is usually annoying.

As can easily be noticed by looking at Figures 6 and 7 both layouts are visually poor and use high contrast colours to ease the process of point selection. The main difference is the amount of interface elements displayed

at the same time, which results in a lower selection throughput for the visible light tracking layout.

The complete interaction pattern implemented by the control application can be subdivided in two main components referred to as active and passive interface. The former takes place when the user wants to explicitly issue a "command" to the house environment. Such a command can either be an actuation command (open the door, play the cd, etc.) or a query command (is the fridge on?, ...).

The second part, instead is related to alert messages or actions forwarded by the House Manager and the Interaction Manager for the general perception of the house status. Alerts and actions must be managed so that the user can timely notice what is happening and provide the proper responses. They are passive from the user point of view since the user is not required to actively perform a "check" operation, polling the house for possibly threatening situations or for detecting automatic actions. Instead, the system pro-activity takes care of them. House state perception shall be passive as the user cannot query every installed device to monitor the current home status. As in the alert case, the control application shall provide a means for notifying the user about state changes in the domestic ambient.

The alerting mechanism is priority-based: in normal operating conditions, status information is displayed on a scrolling banner, similar to those of TV newscasts. The banner is carefully positioned on the periphery of the visual interface avoiding to capture user's attention too much and is kept

out of the selectable area of the screen to avoid so-called "Midas Touch" problems [17] where every element fixed by the user gets selected. In addition, the availability of a well known rest position for the eyes, to fix, is a tangible value added for the interface, which can therefore support user pauses, and, at the same time, maximize the provided evironment information. Every 20 seconds a complete check cycle warns the user about the status of all the home devices, in a low priority fashion.

Whenever a high priority information (alerts and Rule Engine actions) has to be conveyed to the user, the banner gets highlighted and the control application plays a well known alert sound that requires immediate user attention. In such a case, the tracking slowness can sometimes prevent the user taking the proper action in time. So, the banner has been designed to automatically enlarge its size on alerts, and to only provide two possible responses (yes or no) for critical actions. As only two areas must be discriminated, the selection speed is sensibly increased and, in almost all cases the user can timely respond to the evolving situation.

5.2 Direct interaction

When the objects to be controlled or actuated are simple enough, a direct interaction approach can avoid the drawbacks of a conventional environmental control system that typically utilises eye interaction with representative icons displayed on a 2D computer screen. In order to maximize the interface efficiency in these cases, a novel approach using direct eye interaction

with real objects (environmental devices) in the 3D world has been developed. Looking directly at the object that the user wishes to control is an extremely intuitive form of user interaction and by employing this approach the system does not inherently need the user to sit incessantly before a computer monitor. This then makes it suitable for implementation in a wider range of situations and by users with a variety of abilities. For example, it immediately removes the need for the user first to be able to distinguish small icons or words, representative of environmental controllable devices, on a monitor before making a selection.

The approach is termed ART Attention Responsive Technology [18]. For many individuals with a disability the ability to control environmental devices without the help of a family member or carer is important as it increases their independence. ART allows anyone who can control their saccadic eye movements to be able to operate devices easily. A second advantage of the ART approach is that it simplifies the operation of such devices by removing the need to always present the user with an array of all potential controllable environmental devices every time the user wishes to operate one device. ART only presents the user with interface options directly related to a specific environmental device, that device being the one that the user has looked at.

5.2.1 Attention Responsive Technology (ART)   With the ART approach the user can sit or stand anywhere in the environment and indeed move about the environment quite freely. If s/he wants to change an environmen-

tal devices status, for instance to switch on a light, the user simply visually attends to (looks at) the light briefly. The ART system constantly monitors the users eye movements and ascertains the allocation of visual attention within the environment, determining whether the users gaze falls on any controllable device. The devices are imaged by a computer vision system, which identifies and locates any pre-known device falling within the users point of gaze. If a device is identified as being gazed at, then the system presents a simple dialogue to ask the user to confirm his/her intention. The actual interface dialogue can be of any form, for instance a touch sensitive screen or any tailor-made approach depending on the requirements of the disabled users. Finally the user would execute an appropriate control to operate the device.

5.2.2 ART development with a head-mounted eye tracker A laboratory-based prototype system and its software control interface have been developed [19, 20]. To record a users saccadic eye movements, a head-mounted ASL 501 eye tracker (http://www.a-s-l.com/) is used as shown in Figure 8. This comprises a control unit and a headband, on which both a compact eye camera, which images one eye of the user, and a scene camera, which images the environment in front of the user, are mounted. Eye movement data are recorded at 50Hz from which fixation points of varying time periods can be derived. In order to calibrate the eye movement recording system appropriately the user dons the ASL system and then must first look at a calibration chart comprising a series of known spatially arrayed points. The

**Fig. 8** ASL 501 headband attaching the two optics system.

relationship between the eye gaze data from the eye camera and their corresponding positions in the scene camera are built up by projecting the same physical point in both coordinate systems using an affine transformation. Eye data are therefore related to the scene camera image.

In order for the ART system to recognise an object in the environment all controllable devices are first imaged by the system. To do this each device is presented to the scene camera and imaged at regularly spaced angles when their image SIFT features [21] are extracted. These features are then stored in a database. New devices can easily be added, as these simply need to be imaged by the ART system and their SIFT features automatically added to the database. To complement each device added, the available device control operations for it are added to the system so that when that device is recognised by the ART system then such controls are proffered to the user.

In order to operate a device the user gazes steadily at the device in question. The ART system recognises the steady gaze behaviour (the time parameter of this fixation can be user-specified), the users eye gaze behaviour is recorded and a stabilised point of gaze in 3D space is determined as shown in Figure 9(a). This gaze location information is then analysed with respect to the scene camera image to determine whether or not it falls on any controllable object of interest. Figure 9(b) shows the detection of such a purposeful gaze. A simple interface dialogue, as illustrated in Figure 9(c), then appears (in the laboratory prototype this is on a computer display) asking for the user to make his/her control input and the system then implements the control action necessary.

There are two parts to this control interface; the information and feedback offered to the user and the input that the user can make to the system. The former is currently a computer display but could easily be something else, such as a head-down display or audio menu rather than a visual display. The input function can also comprise tailor-designed inputs e.g. touchable switches, chin controlled joy stick, sip/puff switch, or by gaze dwell time on the displays buttons, depending on the capabilities of the disabled user. In the first ART development the actual device operation was controlled by an implementation of the X10 protocol, in this work, instead, the ART system has been connected to the House Manager, enabling users to issue commands to almost every device available in their homes, without being bound to adopt a specific domotic infrastructure.
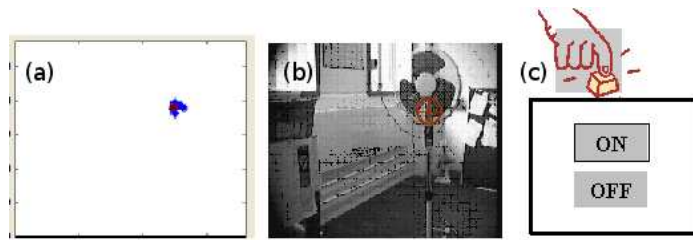
Fig. 9 Typical stages of the ART system (a. Stability of eye gaze captured b. Gaze on object detected c. Control initiated)

One issue of an eye controlled system is the potential false operation of a device simply because the users gaze is recorded as falling upon it. Inherently the users gaze must always fall on something in the environment. There are two built-in system parameters to overcome this. Firstly, the user must actively gaze at an object for a pre-determined time period; this is both necessary for the software to identify the object in the scene camera image as well as preventing the constant attempts by the ART system at identifying objects unnecessarily. Secondly, the users eye gaze does not (of itself) initiate device operation but instead initiates the presentation of a dedicated interface just for that device. This permits a check on whether or not the user does in fact wish to operate the device. The ART system work flow is illustrated in Figure 10.

6 Conclusions

This paper presented a comprehensive approach to user-home interaction through gaze able, on one side, to interface whatever domotic network or de-
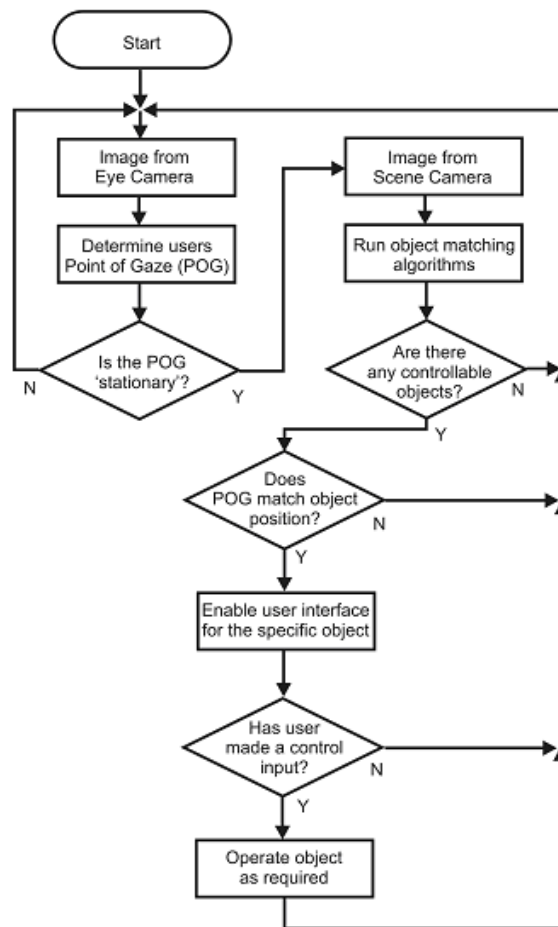
Fig. 10 ART system flow chart

vice with a communication interface, and on the other side to provide several

interfacing mechanisms that can be easily adapted to both user needs and

device complexity. Two interaction patterns have been explored in more

deep detail: direct interaction and mediated interaction. The two, rather

than being used one in opposition to the other, have been integrated mix-

ing the simplicity of direct interaction with the flexibility of PC-mediated

interfaces. The resulting architecture promises to be quite effective in helping disabled users and elderly people to autonomously live in their homes for a longer time.

References

1. The Konnex association. `http://www.konnex-knx.com`.

2. The My Open BTicino community. `http://www.myopen-bticino.it/`.

3. X10. `http://www.x10.com`.

4. The LonWorks platform. `http://www.echelon.com/developers/lonworks/default.htm`.

5. M.A. Just and P.A. Carpenter. Eye fixations and cognitive processes. In *Cognitive Psychology 8*, pages 441–480, 1976.

6. Vertegaal, A. Mamuji, C. Sohn and D. Cheng. Media eyepliances: using eye tracking for remote control focus selection of appliances. In *In CHI Extended Abstracts*, pages 1861–1864, 2005.

7. J. Wang and E. Sung. Study on eye gaze estimation. In *IEEE Transactions on Systems, Man and Cybernetics, part B: Cybernetics*, volume 32, pages 332–350, April 2002.

8. L. R. Young and D. Sheena. Survey of eye movement recording methods. In *Beh. Res. Methods Instrum., vol. 7, no. 5*, pages 397–429, 1974.

9. R. Bates and O. Spakov. Implementation of COGAIN Gaze Tracking Standards. In *Deliverable 2.3, COGAIN Project*, 2006.

10. L. Jiang, D. Liu, B. Yang. Smart home research. In *Proceedings of the Third Conference on Machine Learning and Cybernetics SHANGHAI*, pages 659–664, August 2004.

11. The BTicino MyHome system. `http://www.myhome-bticino.it`.

12. F.Shi, A. Gale, K. Purdy. Direct Gaze-Based Environmental Controls. In *The 2nd Conference on Communication by Gaze Interaction*, pages 36–41, 2006.

13. D. Bonino and A. Garbo. An Accessible Control Application for Domotic Environments. In *First International Conference on Ambient Intelligence Developments*, pages 11–27, 2006.

14. P. Pellegrino, D. Bonino, F. Corno. Domotic House Gateway. In *Proceedings of SAC 2006, ACM Symposium on Applied Computing*, Dijon, France, April 23-27 2006.

15. Francesco Furfari, Lorenzo Sommaruga, Claudia Soria, and Roberto Fresco. DomoML: the definition of a standard markup for interoperability of human home interactions. In *EUSAI '04: Proceedings of the 2nd European Union symposium on Ambient intelligence*, pages 41–44, New York, NY, USA, 2004. ACM Press.

16. OSGi alliance. http://www.osgi.org/.

17. Jacob R.J.K, Karn K.S. Eye Tracking in human computer interaction and usability research: Ready to deliver the promises. In *The Mind's Eye: Cognitive and Applied Aspects of Eye Movement Research*, pages 573–605, 2003.

18. Gale A.G. The Ergonomics of Attention Responsive Technology., 2005.

19. Shi F., Gale A.G., Purdy K.J. Eye-centric ICT control. In *Contemporary Ergonomics 2006, (Taylor and Francis, London)*, pages 215–218, 2006.

20. Shi F., Gale A.G., Purdy K.J. Helping People with ICT Device Control by Eye Gaze, 2006.

21. Lowe D.G. Distinctive Image Features from Scale-Invariant Keypoints. In *International Journal of Computer Vision.*, volume 2, pages 91–110, 2004.