IST-2003-511598 (NoE)

COGAIN

Communication by Gaze Interaction

Network of Excellence
Information Society Technologies

D2.5 Draft standards for gaze based environmental control

Due date of deliverable: 31.08.2007
Actual submission date: 15.10.2007

Start date of project: 1.9.2004 Duration: 60 months

Politecnico di Torino / De Montfort University

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	x
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Main authors: Fulvio Corno (POLITO)
Emiliano Castellina (POLITO)
Richard Bates (DMU)
Päivi Majaranta (UTA)

Main contributors: Howell Istance (DMU)
Mick Donegan (ACE)

Table of Contents

List of Figures

Executive Summary

This deliverable set out to define draft standards for gaze-based environmental control. The deliverable sets out the need for these systems to be both enabling and also reducing the workload of the users of these systems.

Definitions of environmental control systems and smart homes and the requirements of these systems to reduce user workload and increase user autonomy are examined. Existing recommendations for accessible control interfaces are examined, and the current recommendations of European working groups for accessible smart home interfaces are summarised together with the current COGAIN recommendations for gaze driven interfaces in general, with an emphasis on those that may be used for domotic control. Sets of tables reviewing and examining existing systems and interfaces in the context of existing guidelines are given and show that no current systems achieve close compliance with these guidelines for accessibility and control.

The underlying architecture of environmental control systems are reviewed and a new proposed COGAIN architecture suitable for gaze control is given. Existing environmental control system interfaces are examined and critiqued in the light of existing recommendations.

A problem analysis of what data is exchanged in a domotic system, and how the domotic devices and appliances exchange information and how this information is processed is given. This concentrates on the functioning of the system and builds to hypothetical example cases of disabled users operating environmental control systems via these existing systems and existing information flows. The use cases highlight areas where good practice and bad practice are found in the operation.

The deliverable shows that although dedicated gaze-driven systems are available, these do not follow all of the existing guidelines, and that no existing systems are particularly suitable for gaze control when that control is used for flexible and interoperable systems. The application requirements of a gaze-driven environmental control system and interface design, with an emphasis on safety issues and controllability via gaze, are given and a new architecture and a new set of design guidelines that will now be followed and implemented within COGAIN, to give a fully compliant gaze-driven domotic system for users, are shown.

1 Introduction

1.1 This Deliverable

This deliverable presents initial draft standards for the operation and internal structure of gaze-driven environmental controls systems.

The deliverable starts by recapping the definitions of environmental control systems and smart homes and highlights the requirements of these systems to reduce user workload and increase user autonomy. In Section 2, the deliverable then reviews existing recommendations for accessible control interfaces, examining the current recommendations of European working groups for accessible smart home interfaces. The section also examines the current COGAIN recommendations for gaze driven interfaces in general, with an emphasis on those that may be used for domotic control, and gives a summary table of recommendations. The Section then goes on to review the underlying architecture of environmental control systems to place a new proposed COGAIN architecture suitable for gaze control in context. This new architecture is then explained. Finally Section 2 examines existing environmental control system interfaces and offers a critique of the shortfalls of these systems in the light of existing recommendations.

Section 3 gives a problem analysis of what data is exchanged in a domotic system, and how the domotic devices and appliances exchange information and how this information is processed. This section concentrates on the functioning of the system and builds to hypothetical example cases of disabled users operating environmental control systems via these existing systems and existing information flows. The use cases highlight areas where good practice and bad practice are found in the operation.

Section 4, the final section, brings together the application requirements of a gaze-driven environmental control system from the findings of the previous sections. It offers guidelines for system and interface design, with an emphasis on safety issues and controllability via gaze.

1.2 What is Environmental Control?

Environmental control is the control, operation, and monitoring of an environment via intermediary technology such as a computer. Typically, this means control of a domestic home. Within the scope of COGAIN and this deliverable, this environmental control concerns the control of the personal environment of a person (with or without a disability). This defines environmental control as the control of a home or domestic setting and those objects that are within that setting. Thus, we may say that environmental control systems enable anyone to operate a wide range of domestic appliances and other vital functions in the home by remote control.

In recent years, the problem of self-sufficiency for older people and people with a disability has attracted increasing attention and resources. The search for new solutions that can guarantee greater autonomy and a better quality of life has begun to exploit easily available state-of-the-art technology. Personal environmental control can be considered a comprehensive and effective aid, adaptable to the functional possibilities of the user and to their desired actions.

We may say that the main aim of a smart home environment is to:

- **Reduce the day to day home operation workload of the occupant**

1.3 Terminology

Several terms are currently used to describe domestic environmental control. Often a domestic environment that is controlled remotely is referred to as a 'Smart Home', 'SmartHouse', 'Intelligent Home' or a 'Domotic Home', where 'Domotics' is the application of computer and robot technologies to domestic appliances. This is a portmanteau word formed from *domus* (Latin, meaning home) and informatics. All of these terms may be used interchangeably. Within this deliverable, we consider environmental control within the home as a form of assistive technology to aid users with disabilities. In this context we will use the term '*Smart Home*' (unless a system referred to has a different name) as it encompasses all of these terms in a clear and understandable way, and allows for levels of automation and possible artificial intelligence over and above simple direct environmental control of objects within the home.

Often the term 'remotely controlled' gives the impression that the home is controlled from some other place outside of the home (perhaps from the workplace for example). This can be the case, but within the terms of COGAIN 'remotely controlled' means an object or function of the home that is controlled without the need to handle or touch that object. In these terms, the object may be right in front of a user, with that user controlling the object remotely via a computer screen rather than actually handling, lifting and manipulating the object – physical actions that user the user may not be able to accomplish.

1.4 Disability and Environmental Control

When a user has a physical disability, that user may not be able to manipulate physically objects in their environment at all. Thus an environmental control system moves from being a useful labour saving device to a personal necessity for independent living, by enhancing and extending the abilities of a disabled user and allowing independence to be maintained. The environmental control system may be the sole and only way for such a person to control their environment.

Hence, extending the definition of the previous section, when that occupant has a physical disability, the aim of smart home systems is extended, it will:

- **Reduce the day to day home operation workload of the occupant**
- **AND enable the occupant of the home to live autonomously as much as is possible**

Such personal autonomy over their environment has the benefit of reducing the reliance on the continuous help of a carer and/or family member, and increasing the self-esteem of the user as they can control the World around them.

2 Background

2.1 Overview of existing recommendations for accessible Environmental Control Interfaces

Section 2.1 presents an overview of existing recommendations for accessible environmental control interfaces with a focus on recommendations that are applicable to gaze-based systems. It represents a survey of the current state-of-the-art in accessible environmental control interfaces.

Section 2.1 first analyses and summarises recommendations from two standard groups:

- the smart house standards steering group (SHSSG)
- the design for all and assistive technologies standards coordination group (DATSCG).

Both groups were established by the ICTSB the European Committee for Electrotechnical Standardization (CENELEC) and the European Commission DG in to make recommendations to the standards bodies, industry and regulatory authorities on smart home systems and design for all approaches to ICT. The SHSSG recommendations are not directed particularly at accessible environmental control interfaces, but rather at a whole range of smart home facilities; in addition, the DATSCG recommendations cover a wider range of applications over and above smart homes. However, the section summarises the applicable parts of the recommendations. Finally, the section surveys the findings to date, of COGAIN with respect to accessible interface design. The section ends with a summary table of existing recommendations were accessible environmental control interfaces.

2.1.1 Smart House Standards Steering Group recommendations

The SHSSG (http://www.ictsb.org/SHSSG_home.htm) has discussed all aspects of smart house standards. This section however, deals with the aspects related to accessible environmental control interfaces.

SHSSG, in conjunction with ANEC (the European consumer voice in standardisation, a consumer driven organisation that defends consumer interests in the process of standardisation and certification, www.anec.org) have identified the key areas that uses are concerned with, these are:

- User satisfaction
- Reliability and quality of service
- User interface
- Usability and ease of use
- Design for all and personalisation
- Security and safety

We can now address these issues in terms of the interface and accessibility requirements for the user:

- *User satisfaction* - The satisfaction of the user and the perceived reliability privacy and security of the smart home system are essential. Therefore, it is important that understanding the issues concerning the user are considered to all sections of the design of a smart home. From the user's perspective, smart home represents more than just a high-tech gadget. It represents a long-term commitment and investment into their living and lifestyle, and greatly affects how they conduct their lives.

- *Reliability* - when things go wrong: in any complex electronic and software-based system there is possibility of things going wrong, either by equipment failure or software failure or human error or even malicious damage. Users feel confident, if they know the system has self-diagnosis and reporting of problems and status. These are particularly true when the user is disabled and may not be able to fix or operate the system when it is not working correctly. It is important that the system remains reliable as possible to ensure the user is confident with the system.

- *User Interface* - smart homes will inevitably have a number of different types of use interfaces such as keyboards, switches, touch screens etc, depending on the needs of the user and the type and operation of the system. Each of these systems may have a different method interaction, and this requires the use it either be able to complete this type of interaction, which may not always be possible. This is particularly true with users with a disability may not be able at all to use certain interfaces. In addition, the user should be able to customise or personalised interface to suit their needs, and the best way of understanding the system. This is particularly true when systems are large and complex. The interfaces should not be overly complicated, and interfaces should be our tolerant (this is particularly true with gaze interaction). The user should be able to recover quickly and simply from any accidental or deliberate incorrect commands. A positive feedback or indication should all be given the command has been activated so that the user is confident that that action has taken place. The user interface should be simple to use and navigate and the user should be able to obtain various functions clearly and intuitively and unambiguously. Function should be organised and associated according to their nature, and the way we use them in the environment, so for example, all of the available controls within a given room should be clustered together, or for example, all heating controls could be clustered together on the interface.
- It is important that user interfaces have interoperability so that any given interface of choice from the user can operate any given smart home system. This would give a commonality of approach allowing users to learn any one system or interface would be able to operate many environmental control home systems.

- *Usability* - usability is the relationship between the tools and a users in order foothold be effective it must do what the user what is it to do in the best way is possible. Regarding the user interface, usability may be termed in how well the functionality of the interface fits the user's needs and expectations, how well the flow through the operation of the user interface fits the mental model of the user, and how the response of the user interface fits the expectations of the user. The usability of the interface is the property of it being easy to learn and operate, so for example, the interface may show graphical representations of batons, indicating that these are things which are pressed to operate a function, or it may have graphical representations of indicator lights showing that the operation is either on or off.
- Navigation must be intuitive and simple. Where the interface has more than one level of navigation, that is where one display leads to another, the way that this must be accomplished should be easy to understand and should not require large amounts of memory or cognition to know where one is in the hierarchy (this is particularly true for gaze based interaction where the use of large buttons may require several layers or hierarchies of interaction to cover many commands). Reversal or backing out the hierarchy should be simple and intuitive.

- Usability goals should extend such that any person seeing a piece of equipment represented on the user interface for the first time should be able to understand what it is and operate it. This must include the ability to easily start and stop that action.

- *Design for all* - one of the objectives of Europe is the integration of people with disabilities into the information society. This includes smart homes. With respect to the user interface of a smart home, this would require a user interface, which is easily adaptable to different users and their needs, for instance by being customisable, and also the use of interfaces which are standardised and are capable of being accessed by specialist new interaction devices (within the context of this report this would be gaze-based interaction). Within these requirements, the user interface should offer more than one way or mentality of operation for any given command or feedback. For example, by using both a sound and a visual indicator to notify the user of an error, or by using both text and visual labels simultaneously, or by allowing the user to issue commands to the system by typing of selecting something with a pointer or by scanning, or other alternatives include methods. Accessible user interfaces and design for all should aim not to try to design for the middle ground or average user, but to design for a broad range of audiences. This may even mean designing separate interfaces were different user populations.
- *Disability access* should be golf all systems enabling people with disabilities to effectively use the systems. This may include for people with motor impairments such things as sticky Keys and slow Keys hardware devices such as head mounted input devices and possibly eye tracking systems. For uses with the impairments this would extend to screen enlargement utilities and auditory output and possibly textual output and text-to-speech systems. Full hearing impairments this would be visual alerts speech to text captioning, and for finally cognitive impairments this may be simplicity of systems and reminder-based systems.

- *Safety and security* – the smart home system should be checking for safety with failsafe guidelines failsafe is a property of a device that inherently prevents certain failure modes or hazards. Although this is not necessarily specifically a concern of the user interface. The user interface should reflect the safety aspects of the system, and the user interface should clearly alert the user to any safety issues. A manual override facility should be possible. The system should provide feedback and vulnerable users abusing the system. This feedback should be directed at carers in the event of system failures of emergencies. This may extend to automatic calling out of security or health care providers or carers.

2.1.1.1 SHSSG standards progress in environmental control interfaces

SHSSG belongs to the CENELEC TC205 standardisation group that addresses smart home standards. Figure 1 shows the various areas that are addressed by this group, it is noticeable that very few of these areas are concerned with accessibility or safety related issues, which are very much the subject of this document. However, it is encouraging that functional safety and safety related issues are to be addressed in future with EN50090-2-4 (see top right of Figure 1).

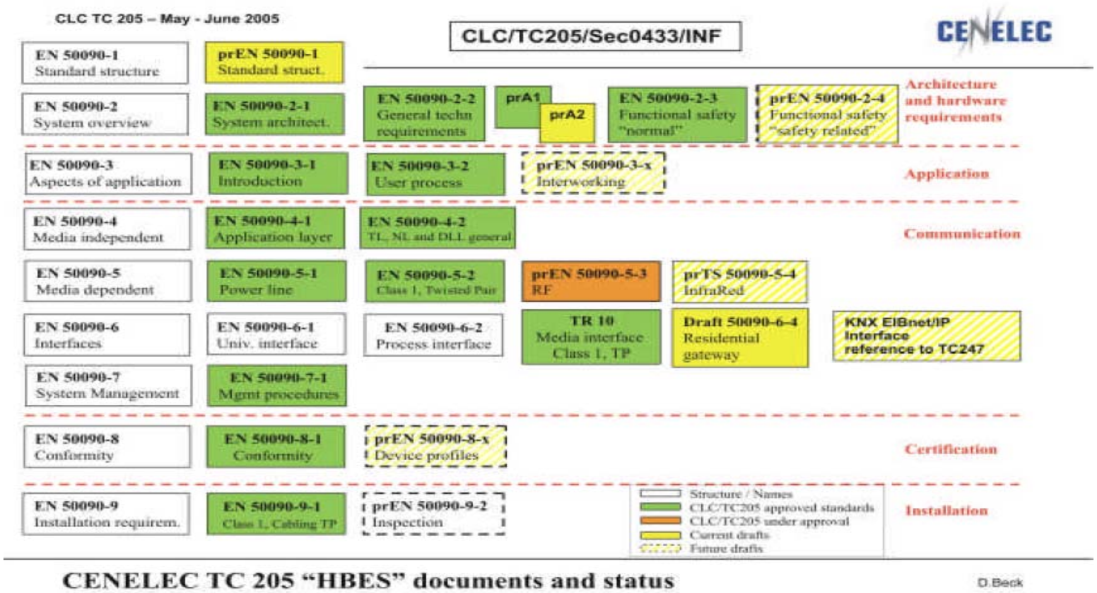

Figure 1. The Status of TC 205 standardisation (05/11/2005)

The SmartHouse Project belongs to TC 205 Home and Business Systems (WG16).
The range of Home Systems Standards and their progress is shown above.

2.1.2 Design for all and assistive technologies standards, coordination Group recommendations

DATSCG (http://www.ictsb.org/activities_Design.htm) has discussed smart phone design and accessibility as part of larger reports on other ICT areas. This section discusses their findings and recommendations based around the user interface of smart homes.

- *General universal design issues* - DATSCG have recognized that smart house systems can be very useful to many user groups and can give the potential to adapt a house to changes in the residence needs and functional capacities. However, DATSCG note that it are still very few smart home installations in ordinary dwellings, but that these installations offer elderly people or people with disabilities greater safety and independence. Disability is of note include changes in vision hearing sense of touch mobility reaction times and memory, cognitive ability, and with increased fear of accidents and problems of living alone. There is a recognized fear of accident or isolation. It is extremely important to adopt design for all and universal design principles within smart homes.

- *The physical environment within the smart home* - elements within the home must be placed within easy access of the people who occupy the home bearing in mind their physical capabilities and mobility as well as auditory visual and cognitive needs. This includes the need to place objects that are being controlled and which are hazardous in some form within the field of view of the user when they are operating them. Failsafe features must be provided.

- *The user interface* - a standardized interface must be developed to input devices across smart house systems with convergence of different modes of operation. Within the interface all interaction elements must be easily detectable and recognizable and they must be logically distributed in terms of the process. The display resolution and illumination should be adequate. Standardized symbols, for example for the state of the device such as on and off should be present on the interface in the should be logical and understandable. Feedback should both be visible and auditory. The interface should be simple and intuitive to use and should give a tolerance for error when operating the home. This tolerance is as important as the implications of errors can be serious in an automated environment. Within the interface, the interface should be customizable to accommodate different mental models of the home. In some cases, the user may not be entirely capable of operating the full function of the home and in these cases the system must be intelligent enough to do it for them when necessary.

- *Ethics* - An ethical approach should be adopted so that any automated safety systems, all reporting of the well-being of the user should be determined beforehand and be in the best interests of the user.

- *Feedback* - The user interface should offer the possibility of multimodal feedback such as acoustic tactile and visible feedback. Continuous feedback should be present on the interface during operation, to enable the user to fully understand the state of the home.

- *Consistency* - The user interface must be consistent with the expectation and intuition of the user, so for example information should be grouped with a consistent method.

- *Adaptation* - The interface should adapt to the different uses needs and desires. For example for users with differing physical control abilities, this could mean the system operating either under full manual control, or for automatic control or some position between the two states. This could result in the home operating automatically, with the house being so smart that it looks after the occupant. Under these circumstances, ethical guidelines the implications in the cases of surveillance and safety must be addressed.

- *Security of operation* - the system must be failsafe and easy to learn. Confusion can lead to insert your operation and it is particularly poor that the system is easy to understand and area safe. Any controls they have a dangerous action or implication must be kept within safe control parameters, such as requiring affirmation of the desired command. Hazardous elements must be eliminated isolated or shielded from the user.

- *Automation* – The smart house system conducts some functions automatically and without action from the user. An example might be a fire alarm automatically calling for help, and then opening all entrances and switching on lights on to make escaping easier. And perhaps also sending a message to home care service. Another example might simply be to switch off lights in unoccupied rooms automatically. In scenarios such as the first example, where considerable level of actions take place automatically and have considerable implications, there must be some prior agreement with the user and healthcare providers and other interested parties on the operation of these automatic features.

2.1.3 COGAIN interface recommendations from other work packages

This section shows the findings already determined by COGAIN that can be applied to interface design for accessible environmental control applications.

2.1.3.1 WP3 and User requirements of the interface

Examining the outputs and findings of work package 3 (Donegan, M. et al. (2006) ***D3.3 Report of User Trials and Usability Studies***. Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 3.3, available at http://www.cogain.org/results/reports/COGAIN-D3.3.pdf), the following statements may be made about the interface design for gaze based systems, including those which would control and environment:

- *Choice of interface* - It is essential to give the user control over their choice of interface. Every user, particularly with a disability has different requirements of any given news interface, and so the widest possible variety of on-screen software interfaces for an environment control system would need to be offered. This would expand the range of choices and the likelihood of matching the interface to the uses needs and abilities.

- *Interface object sizes* - The system should be designed to accommodate inaccurate or difficult pointing using an eye tracking system. Thus, interfaces should be designed with appropriately sized targets. Despite the fact that some users may be able to select small targets interface should also be able to present larger targets as these are easy to select and often require less workload. This is illustrated in Figure 2, where although a 2 x 8 grid can be used by the user. They often opt for a 2 x 2 grid to give sufficiently large targets to enable very easy operation under any circumstances.

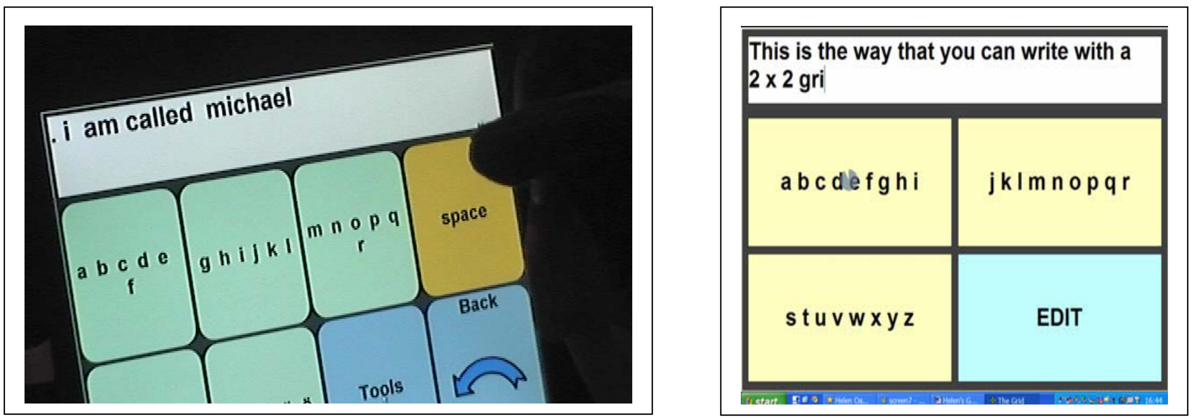

Figure 2. Example interface object sizes

Despite the fact that a user was able to write using the more efficient grid set (left), he preferred to write using the less efficient grid set (right) because it required less effort

- *Interface object placement* - Users should be given the option of different positioning is of interface targets. Users have been found at any easily move their eyes in one plane, for instance only left-to-right horizontally or up and down vertically. In these cases, the interface controls need to be arranged in either horizontal rows or vertical columns to suit the uses needs. The vertical arrangement is shown in Figure 3.

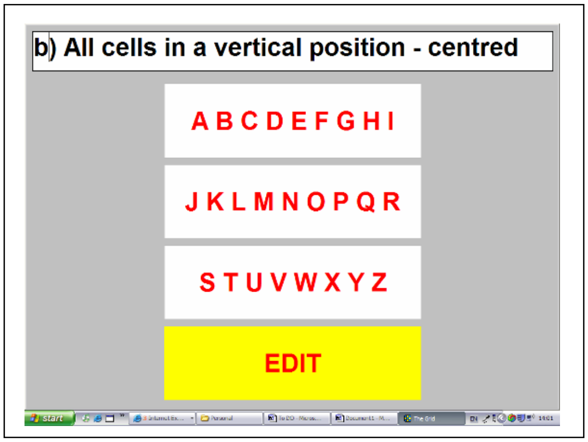

Figure 3. Interface object placement

Russell's writing grid was similar to Michaels 2 x 2 grid (above), but the cells were presented in a way that he could access them using his vertical eye movement.

- *Interface feedback* - Feedback should be given from the interface when the user requires it. Auditory support is most important with gaze based control as it frees the user from looking for feedback from the interface. In addition, however, visual feedback should also be provided to enable the user to see that their command actions have been carried out.
- *Gaze position feedback* - The cursor position should have the option of being shown or not shown to the user. For many uses of gaze systems, it is important that they see where the gaze tracking system is placing the computer cursor driven by the eye. This enables them to be fully aware of where their gaze is tend to be fully aware of what controls they are about to operate. However, for some users the presence of the cursor position can be distracting, and in these cases, the option should be available to make the cursor invisible, with gaze position only shown when the gaze of the user is on a button or object on the interface. In these cases, the feedback would be such that the button or object may highlight show the gaze is on that object. This is shown in Figure 4, where there is no cursor shown at the centre of the object and in this case the selected key is highlighted.
- *Selection feedback* - Selection by gaze should be as unambiguous as possible. In the cases where a user has no second modality such as clicks, which to operate a command, then dwell clicking will be required. In these cases, visual feedback of the dwell time elapsing and expiring should be shown. This is shown in Figure 4, were on the character '**d**' not only is the gaze position is shown, but also the dwell time expiring is shown by the circle filling in a clockwise fashion.

Figure 4. Visual feedback

Screenshot to illustrate the MyTobii dwell-select option for use with cells in grid software.

Examining the findings outputs from work package 4 (Hansen, J.P., Johansen, A.S., Donegan, M., MacKay, J.C., Cowans, P., Kühn, M., Bates, R., Majaranta, P., and Räihä, K.-J. (2005) ***D4.1 Design specifications and guidelines for COGAIN eye-typing systems*.** Communication by Gaze Interaction (COGAIN), IST-2003-511598: Deliverable 4.1, available at http://www.cogain.org/results/reports/COGAIN-D4.1.pdf) the following statements with regard to environmental control interfaces, driven by gaze may be made:

- *Interface object sizes* - Regardless of which interface software is used, it needs to be as flexible as possible to suit the needs of the user. As found in work package 3. This means the ability to change the size and number of interface objects presented to the user. Figures 5 and 6 illustrate to different interfaces used by users. Figure 5 shows that one user preferred full range of a QWERTY keyboard plus accessories be visible at all times even though the interface objects were small and difficult to select by gaze. Figure 6 shows that are different user preferred the interface with four large interaction objects, a hierarchy of objects behind each of the initial objects would be required to offer the user a full range of interface functions. Here we see a trade-off between all the commands available from one click, but with small targets and all of the commands available by many clicks but with large targets. It is important to give the user the choice of which part they wish to follow.

- *Supporting control methods* - A wide range of input methods in addition to eye control should be available to control the interface. This will allow the user to change from one method to another if and when they wish to or their condition dictates, to distribute their interaction effort so they may switch between different control methods during the day, to enable multimodal axis were both gaze tracking plus another modality may be used, and finally to use the interfaces in a wide variety of environments. In this final case, it is found that some gaze tracking systems do not work very well in bright sunlight. So in these cases, users may wish to revert to a different input control system (such as scanning with switch input) when they are outside or very close to a sunny window.

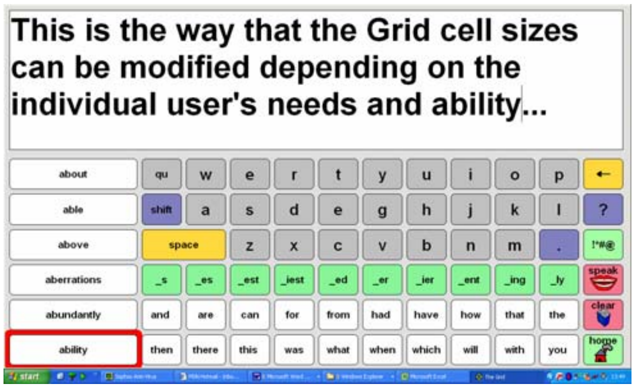

Figure 5. Small object sizes

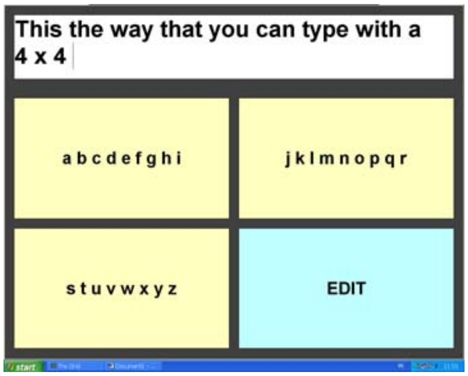

Figure 6. Large object sizes

Both of these grids were designed with the same framework program. The program is flexible enough to be used by those who can eye-point accurately (Figure 5) as well as those who cannot eye-point accurately and need larger cells (Figure 6).

2.1.4 Summary of existing recommendations

The table below gives a summary of the existing recommendations for accessible environmental control system based on the contents of the section:

Recommendation number	Existing recommendation
CENELEC TC 205	• Create consistent user interface • Standardize user interface of smart house systems • Make smart house systems interoperable • Require upward compatibility among different system versions • Create standard choice of different interfaces, individually adapted • Provide failsafe features and manual systems override • Create standards on feedback typologies for all major user groups. • Develop standard requirements for which basic functions a house must have, and where they are placed in the building to provide flexibility
CENELEC TC 205 TC 215	• Require flexibility and modularity • Develop standardised installation aspects • Provide a centrally placed and easily accessible technical centre in the house • The components must be placed within easy access for people with all physical varieties and mobility modes, as well as auditory, visual and cognitive needs. • Standard height for hidden conduits. • Provide sufficient room for extra components and extensions for electricity and ICT in the technical centre • Hide conduits to windows and doors for electricity for control or alarms
CENELEC TC 122	• Create standardised interface to control and input devices across smart house systems and suppliers • Convergence of different modes for operation
CENELEC other TCs	• Standardised symbols e.g. for on-off. • Standardised interface to external display system (e.g. TV, large display). • Adapt standards
COGAIN D3.3 and D4.1	• Resizable cells and grids • A range of input methods • Choice of symbol vs. text input • A selection of text styles and colours • A range of speech facilities • A choice of language • Easy shift between national language and English version • Adjustable speed • Access to system commands from within the eye-writing system

Table 1. Summary of existing recommendations

2.2 Architecture of an eye-tracking based environmental control system

Standardization in domotic systems is still very low, with different vendors, protocols, device families still competing without a clearly defined mainstream solution. Basically, in a complex smart home environment we may have two different categories of intelligent devices (see the right part of Figure 7):

- Devices belonging to some structured domotic network (this is the case of switches or lights connected through a domotic bus such as EIB/KNX or X10), where each device has a limited intelligence and a predefined functionality, and is only able to connect over the specific domotic bus. In this case, the domotic bus is usually equipped with a sort of local gateway, or interface, able to convert domotic commands to/from a TCP/IP protocol, making it easier to integrate with other devices, busses or network in the home. However, the protocols adopted are extremely different and have not been standardized: while TCP/IP solves the connectivity problem, specific knowledge about the implemented protocols is essential to interface with the system.

- Autonomous devices, with higher intelligence, and that are usually able to directly connect to a TCP/IP network. In this report, these devices are called *appliances* to make a clearer distinction. Domotic appliances (such as media centers, game consoles, anti-theft systems, VOIP phones, wireless cameras, …) are increasingly common and, unfortunately, increasingly diverse. Diversity comes both from their supported functionality and from their network control protocol. In such cases each device can normally be controlled by a web browser or by a specifically tailored application, but no interoperability is usually guaranteed.

Due to this diversity of domotic busses, smart appliances, and their control protocols, we have two different types of environmental control applications:

1. *closed* systems, that are able to interface with one type of domotic protocol or infrastructure. Such applications are usually developed with specific development kits supplied by the technology manufacturers. In this case, if we refer to the left part of Figure 7, the User Interface presented by the Access Device will “talk” directly with the domotic Interface (in the case of domotic devices) or with the smart appliance. In such cases usually the access device is the same as the computer running the user interface.

2. *multi-standard* systems, where the control application is able to integrate the control of different types of appliances and of different domotic buses. This is necessary to support complex scenarios, to have versatility of user interface access, and to be able to command a device independently from its physical connection (e.g., switching off the garden lights using the TV remote control). Various approaches have been proposed for this general solution, ranging from completely decentralized to totally hierarchical protocols, but any approach has inevitably to deal with reconciliation of incompatible protocol, support of a wide range of functions, and a varying degree of controllability and observability of each device or appliance. In this second case, one further component is needed, that in the center of Figure 7 is called Gateway, able to handle all the necessary protocols and to connect to every device and appliance, and to present them transparently and in a unified way in the user interface.

In the rest of this section we will try to analyze the most important problem arising in the development of control interfaces for domotic systems. It should be already clear that device interoperability is not guaranteed, and that a good deal of specific competence is needed in order to successfully interface to any one system or appliance. This situation contrasts with the COGAIN goals, where we want to create a simple, comprehensive, and standard path for [eye tracking] application developers to add environmental control features to their applications.

While presenting the problems, we will also outline some solutions, whose general structure may be assumed as a standard way of integration in COGAIN, that were successfully experimented by the project partners.

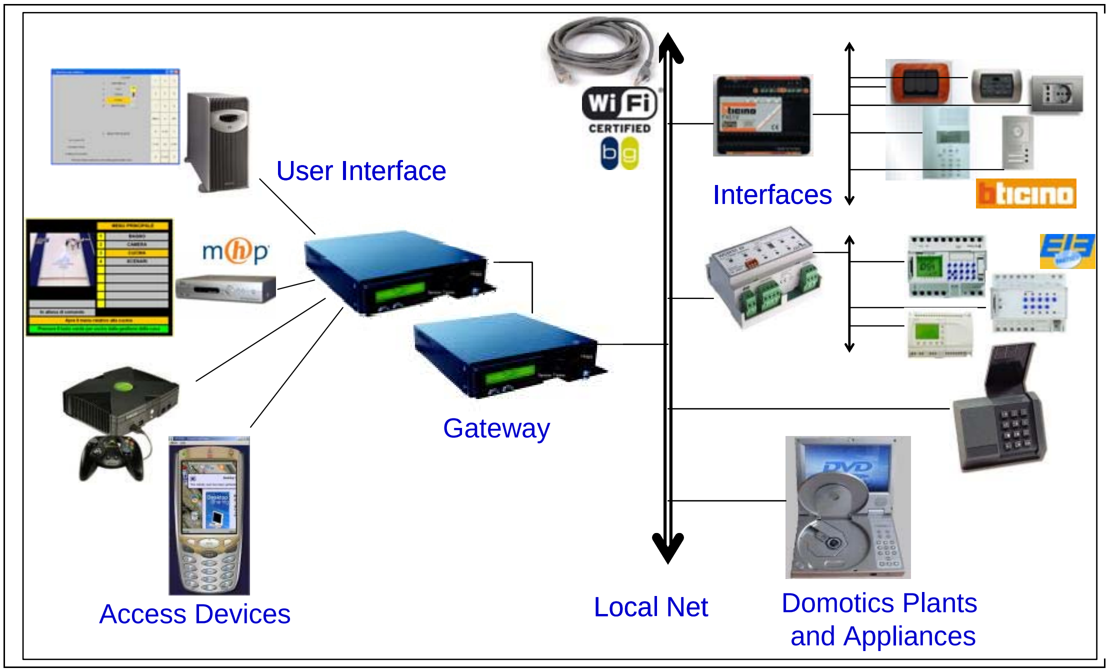

Figure 7. Overview of Domotic Systems

Generic architecture of a complex multi-vendor smart home infrastructure. The roles of the Access Device, the User Interface and the Gateway may be attributed to different devices or they may be concentrated in the same device, according to specific design choices.

2.2.1 A Bridge to Many Diverse Systems

As determined and shown previously in Deliverable 2.4, yhe key to adopting as many systems as possible is for COGAIN to develop a *bridge* between gaze based systems and environmental control systems. Such a bridge would wrap the interfaces and protocols of the many diverse systems found in the survey of D2.4, interacting with each of these systems, and translating their differing commands and data into a single unified COGAIN ‘Smart Home Standard’ protocol that would be compatible with gaze control.

The role of the COGAIN driver can be illustrated by taking the diagram of Figure 8, that shows where the COGAIN bridge is placed between the core domotic system and any gaze driven interfaces.

The following sub-section will lay out the foundations for such a standard.

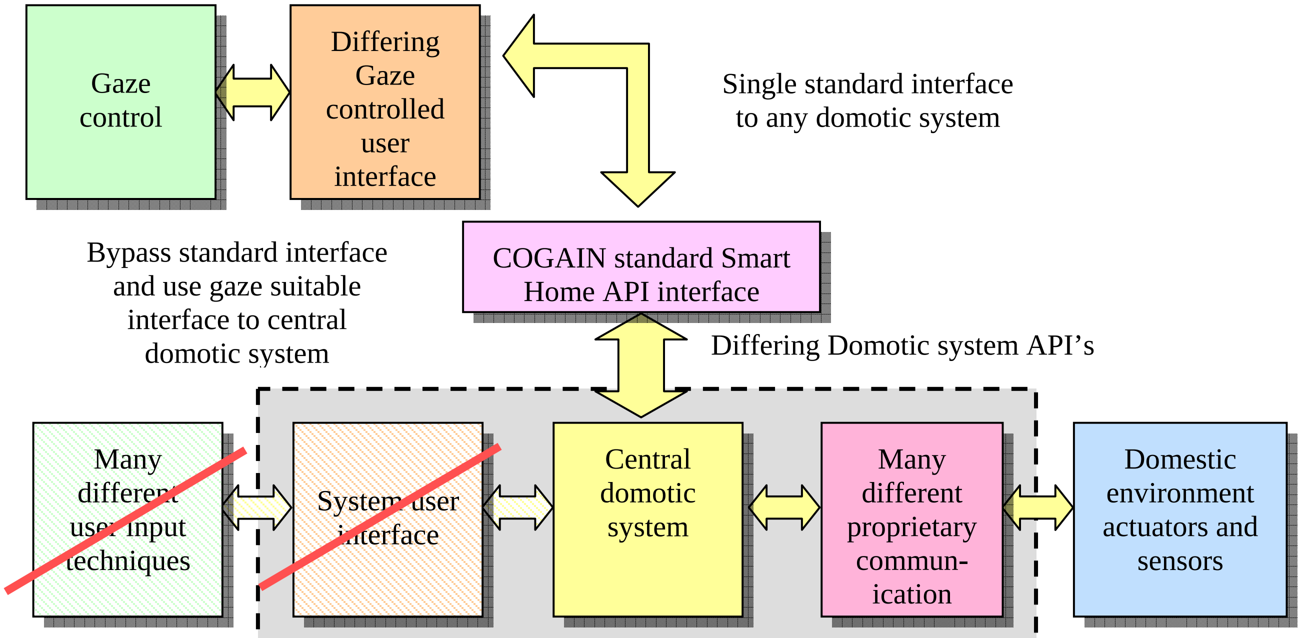

Figure 8. COGAIN Smart Home standard interface

2.2.2 Architectural components

Any complex smart home is integrated by several types of components, cooperating in a multi-layer architecture. A comprehensive picture summarizing the different layers and the main components in each layer is reported in Figure 9.

The four main layers, bottom to top (right-to-left in the figure) are: the Smart Environment layer, the Protocol, Network, Access interface layer, the Control application user interface layer, and the User interaction device layer. The following paragraphs specify in more detail the structure and the role of each of these layers, and its internal devices/functions.

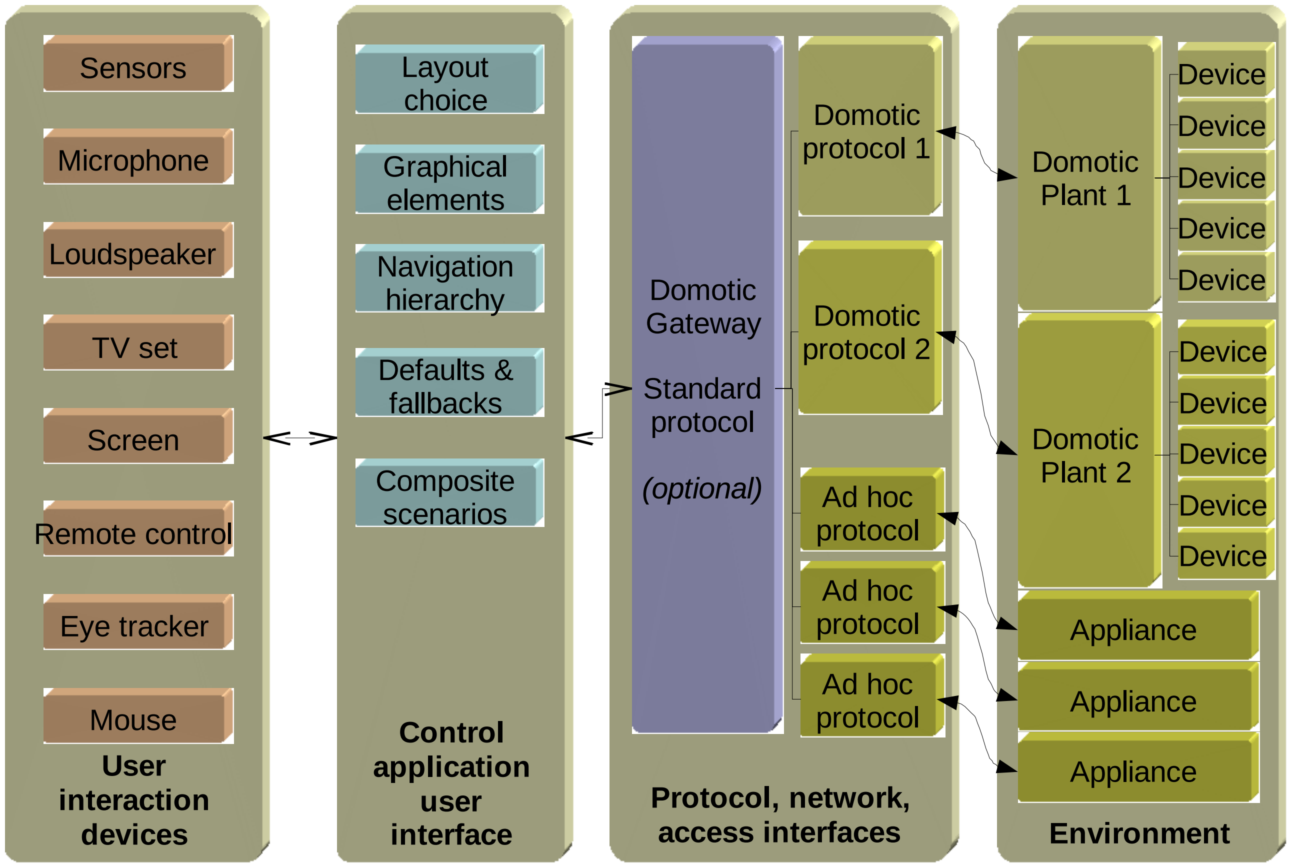

Figure 9. Layers and Components in a generic Smart House

- **Smart Environment layer**. This layer comprises the actual equipment composing the smart house. Such equipment is able to operate independently, autonomously, even without the additional layers (but in this case with limited or no programmability). Such layer consists of zero or more domotic plants, plus zero or more appliances. In particular:

  - **Domotic plants**. Each plant is characterized by a specific technology. The technology comprises aspects such as: bus technology (wired, wireless, powerline, …), bus protocol (usually proprietary), types of supported devices, brands that produce such devices. The plant technology is either defined unilaterally by a single manufacturer, for its products, or is the result of an industrial standard agreed among a group of manufacturers. For an overview of the available plant technologies please see Deliverable D2.4. We assume that each domotic plant is equipped with at least one "intelligent" interface gateway able to convert bus traffic (that is a low-level proprietary issue) into some for of TCP/IP protocol. Such device is not strictly needed, but all advanced plants should have one anyway.

  - **Domotic devices**. Such devices, attached to a plant, are strictly technology dependent. Each plant manufacturer (or manufacturers adhering to some standard) usually offers a wide range of devices: switches, buttons, relays, motors (for opening/closing doors, shutters, windows, …), sensors (presence, intrusion, fire and smoke, temperature, …), as well as a range of multimedia capabilities (ambient microphones, analog or digital cameras, loudspeakers, radio tuners, …). Some plants are also integrated with the communication system in the house (telephone, intercom, …) and sometimes with entertainment systems (TV sets, media centers, radio tuners, …). The simplest devices (i.e., switches, relays, motor controls) offer similar functionality across all manufacturers and plants, even if the addressing and activation details differ. More complex devices tend to be more dissimilar across manufacturers, and therefore are more difficulty to classify in a uniform manner and to integrate seamlessly into control interfaces.

- **Smart appliances**. Thanks to the explosive growth of consumer electronics, enabled by digital media technologies that call for new ways of dealing with photographs, music, and video, "smart" appliances are rapidly increasing both in number and in functionality. Such appliances often aim at covering the communication and multimedia needs of the users. For these kinds of devices, «Current end-to-end solutions that are based on proprietary vertical implementations bring products to market early but have little impact on rapidly establishing a new category of products. Moreover, end users do not have the opportunity to select parts of a system from different manufacturers because there is no interoperability between those non-standard devices. Thus industry leaders must define guidelines to enable an interoperable network of CE [Consumer Electronics], PC and mobile devices.»[1] This problem is starting to be understood by some of the major players in the consumer electronic industry, that teamed to define interoperability standards for *media control* and *media delivery* in the Digital Living Network Alliance (DLNA - http://www.dlna.org/). Such interoperability (Figure 10), even if backed by some industries, is *not yet* a reality. Further, in the context of COGAIN we want to achieve a still higher interoperability level: that of home automation (and not just entertainment) systems, and that of accessible interfaces.

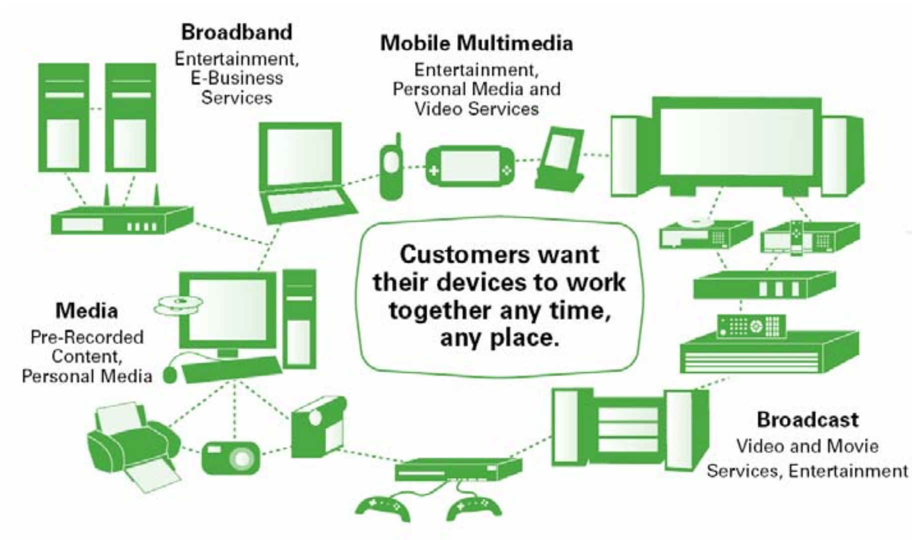

Figure 10. DLNA interoperability vision

Source: DLNA

[1] DLNA Overview and Vision Whitepaper 2007, Digital Living Network Alliance, http://www.dlna.org/en/industry/pressroom/DLNA_white_paper.pdf

- **Protocol, Network, Access interface layer**. If one just wants to operate a smart home system with the functions and the user interface predefined by their manufacturer(s), no additional layer is needed. However, in order to be able to create and customize the user interface, a first requirement is being able to communicate with the smart home. We recall that at the Environment layers there are numerous incompatible protocols to handle. The protocol access layer should comprise the function to:

    - *Interact with the domotic plant(s) according to its network protocol*. This requires first to know the details of such network protocols, which are often proprietary, or are available only commercially. Once the protocol is known, one must properly configure the network (we recall that we are assuming a TCP/IP –wired or wireless– infrastructure in the smart home) and properly authenticate with the system (home security requirements often impose secret or "weird" authentication procedures). Once the connection is established, the protocol handler may issue the commands and enquiries (as requested by the upper layers) to the underlying domotic system. In the cases in which in the same house more than one domotic plant is installed (i.e., you have an anti-theft system of a different brand from your lightning and automation one), several protocols must be supported at the same time.

    - *Interact with the various appliances present in the house*. This requires studying, one by one, such appliances, checking if their support any standard protocol, or reverse-engineering any proprietary communication features they may possess. This is a highly device-specific task, that can be accomplished with varying degrees of success, mainly depending on the openness of mind of the appliance designers. In general terms, we discovered that it is often reasonably easy to *control* a device (i.e., to issue commands that it will execute) rather than to *observe* it (i.e., to query it to determine its current status).

    - *Convert all the protocols to a common one, at the higher level*. Ideally, the control application developer should not bother with all the details and issues of interacting with a complex smart house. In this case it can be useful to design a *common protocol*, encompassing all (or most) functions of lower level protocols, to be understood by a gateway. This component is the basis of the proposed COGAIN standard for accessing smart house environments, as it dramatically eases the work of application developers. The difficulty in the gateway component lays in the complexity and diversity of the information it should exchange (see Section 3.1 for an overview), and the varying level of detail and functionality offered by different protocols on different topics.

- **Control application user interface layer**. Assuming that the previous layer (that may be embedded into the same executable application or may reside on a different server) resolved the connection and protocol issues, we may build control applications for all the intelligent devices and appliances available in the house. Such application, being relieved from low-level control issues, has to concentrate on user interaction issues, such as navigation, layout, and graphics. The control interface for environmental control systems is peculiar in at least three ways, that will be better explored in chapters 3 and 4:

- *Asynchronous additional control sources*. The control interface is not the sole source of commands: the house occupants may choose to operate on wall-mounted switches, or some external events may change the status of some sensors, or the current music track just ended, … In other words, the control interface needs to continuously update its current knowledge about the status of the house (the "model" in the model-view-controller pattern). Icons, menu labels, etc must change according to status evolution, in order to have a coherent view of the environment through its control interface.

- *Time-sensitive behaviour, including safety issues*. Some actions are time dependent, such as the time available to react to an alarm condition. In such cases, the user is put in a stressful condition, since he has limited time to take important decisions, which may pose threats to his safety. In this case the control interface must offer very simple and clear options, easy to select, and must be able to take a correct (i.e., safest) action in case the user cannot answer in time. Another example are actions "initiated by the house" (this means that the rules programmed in the intelligent devices have just triggered some pre-programmed action, such as closing the shutters at nighttime). In this case the user should be allowed to interrupt the automatic action and to override it. The control interface should make the user aware that an automatic action has been initiated, and offer ways to interrupt it (unobtrusively with respect to the current user interaction status).

- *Trade-off between structural and functional views*. From the Information Architecture point of view, we have at least two possible hierarchies by which organizing the user interface: structural and functional. The *structural hierarchy* is the one adopted by most environmental control interfaces, and follows the physical organization of the devices in the house. Successive menus levels represent stages, rooms, devices within the room, and possible actions on those devices. While being natural, since it exploits the knowledge of the house structure, this choice leaves out some important actions: where to put in the hierarchy such "global" or "un-localized" actions as switching the anti-theft system on, or connecting to a news bulletin, or uploading personal health data? Often some "Other", "Scenarios", "Global" or similarly named menu plays the catch-all role for these situations. For this reason, *functional hierarchy* may also be adopted, where actions are grouped according to their nature, rather than by their location.

- **User interaction device layer**. This final layer represents the actual devices (simple as a mouse or complex as an eye tracker) that the user is adopting to interact with the control application user interface. These are usually standard devices or systems, that materialize the tangible part of user interaction. The diversity of these devices, coupled with the desire and need to let the user to select their preferred interaction method, poses additional challenges to the user interface, that must be prepared to prepare different screen resolutions, different pointing precisions, different modalities (e.g. audio instead of graphics), and so on. Interaction between devices and the control interface uses the modalities supported by the device and by the underlying Operating System.

The main goal of this section was to describe the (complex) architecture of a complete domotic system, outlining the main processing layers and the involved components. The key points being stressed were on interoperability and on user flexibility, and we have shown that they are extremely challenging objectives. Some industrial solutions are being sought to interoperability problems, even if only in application niches (house control, multimedia fruition, …) instead of from a user-centric point of view.

During the last years, partners of the COGAIN project worked to find viable (with current technology) solutions that would allow decoupling the complexity of interfacing with domotic systems from the complexity of creating a dynamic and flexible user interface. This decoupling, described in the next section, forms the basis for a proposed **standard COGAIN domotic protocol**.

2.2.3 The proposed COGAIN architecture

In this section an overview of the Domotic OSGi Gateway (DOG), developed by Politecnico di Torino for COGAIN, is presented, followed by a more detailed description of its components. The proposed system has been designed with the intent of supporting intelligent information exchange among heterogeneous domotic systems which would not otherwise natively cooperate in the same household environment. The concept of event is used to exchange messages between a device and the DOG. Low-level events are converted to logical events internally to the DOG so as to clearly separate the actual physical issues from the semantics that lies beyond the devices and their role in the house. In this way, it is also possible to abstract a coherent and human-comprehensible view of the household environment.

2.2.3.1 DOG overview

DOG is realized with OSGi technology that is an Universal Middleware. OSGi technology provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software lifecycle. It provides a general-purpose, secure, and managed Java framework that supports the deployment of extensible and downloadable service applications known as bundles. Every bundle can interact with other bundles just using and supplying services, respecting specification constrains attached to every bundle. In this way it is possible to build up a decentralized engine, which can continuing work even if some of its parts becomes unusable. When a functionality (bundle) becomes available, the DOG is immediately notified and can start using it: the availability of the bundle (and of its services of course) can trigger the availability of other bundles and services. All that, without the necessity to restart DOG.

DOG is composed by different bundles that have various uses and roles. A new bundle is easy to integrate: it only has to conform to the communication constraints defined in the framework.

2.2.3.2 DOG architecture

The general architecture of the DOG is deployed as shown in Figure 11 , and can be divided into three main layers: the layer 1 involves all the devices that can be connected to the DOG and their software drivers, which adapt the device technologies to the gateway requirements. The layer 2 is the main responsible for routing low-level event messages to and from the various devices and the DOG, and also includes the generation of special system events to guarantee platform stability. The layer 2 is also the actual core of intelligence of the system. It is devoted to event management at logical (or semantic) level and to this purpose it includes a rule-based core which can be dynamically adjusted either by the system itself or manually through API interfaces. The rules define the expected reactions to incoming events, which can be either generated by the house, let's say the "door is opening" as an example, or by an application: "open the door".

Complex scenarios may involve the rule engine: for instance, a rule might generate a "switch on the light in room x" event if two events occur: the room x is dark and a sensor revealed the presence of somebody in that room.

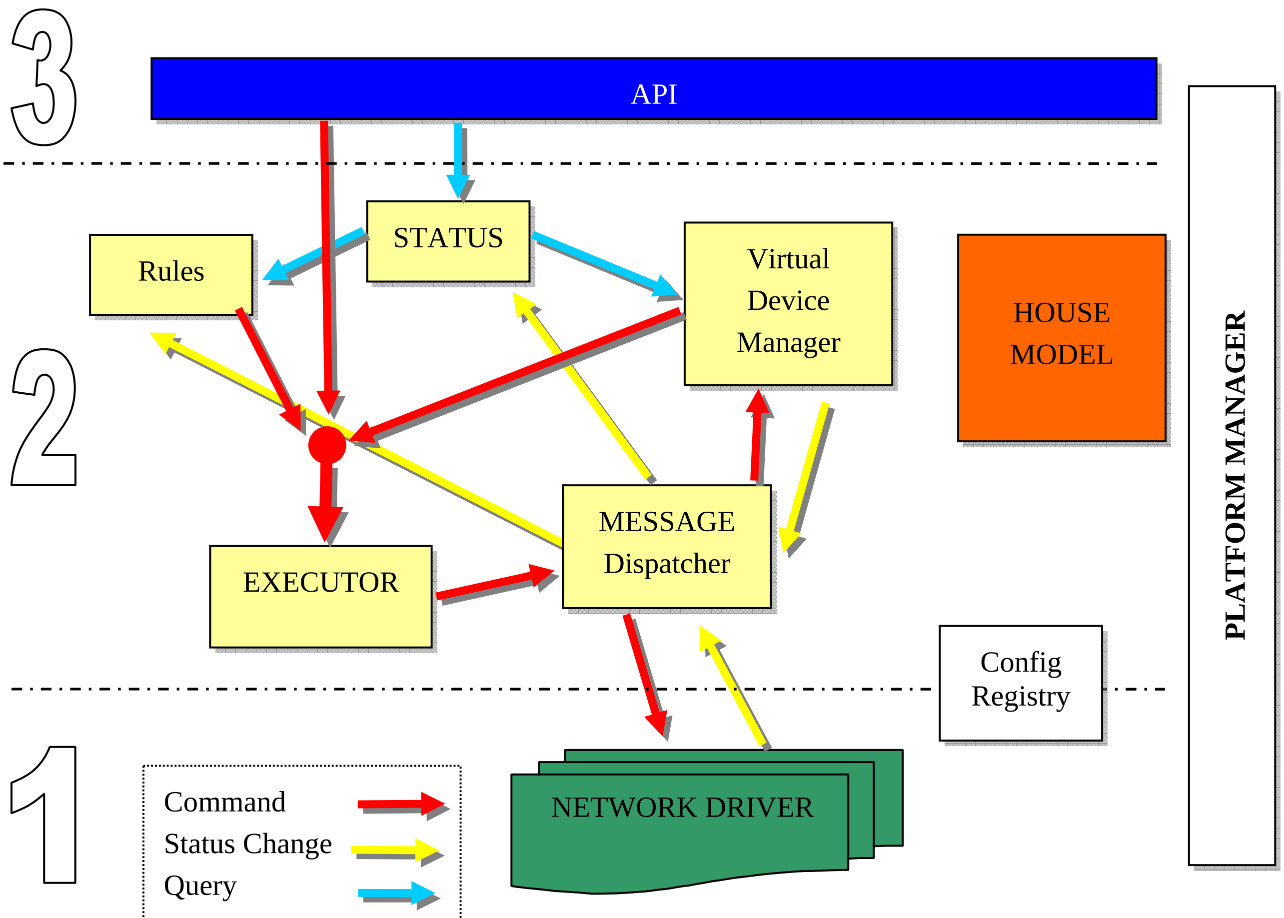

Figure 11 DOG Architecture Internal communication protocol

2.2.3.3 DOG bundles

Figure 11 illustrates the interaction among the basic bundles that compose the DOG.

- **Network Drivers**: these bundles directly interact with the home automation system. It is necessary a different Network Driver for each home automation system low-level protocol. Network Drivers communicate with the Message Dispatcher bundle that easily routes events.
- **Message Dispatcher**: this bundle dispatch events coming from Network Drivers and commands coming from the Executor bundle. Message Dispatcher contains a routing table that map the relation between devices and the Network Driver that can pilot them.
- **Executor**: this bundle listens the commands and requests from Rule Engine, User Applications and Virtual Device Manager the forward them to the Message Dispatcher.
- **Rule engine and Virtual Device manager**: these bundles are the core of the DOG intelligence. The Rule Engine allows the management of user defined scenarios and also automated reaction to alarms. The Virtual Device Manager provides device-level abstraction to real devices adding capabilities through software emulation.
- **House Model**: this bundle contains the representation of the house and devices through the DomoML-env ontology described in DomoML-env: an ontology for Human Home Interaction

(Lorenzo Sommaruga , Antonio Perri, and Francesco Furfari). House Model operates translations from ontology representations and real devices and allows semantic reasoning on the device properties.

- **Status**: this bundle is a cache that contains the status of devices and appliances present in the system. The Status bundle respons to API bundle requests returning information about devices.
- **Config Registry and Platform manager**: these bundles handle the correct start-up of the whole system, giving each bundle the right configuration and parameters.
- **API:** this bundle is the external access point to the system. API bundle provides through methods, the XML-RPC protocol to get house configuration, devices status and send commands. A more detailed description of the API will be available at the beginning of 2008 when the **D**omotic **O**SGi **G**ateway will be released.

2.2.3.4 Internal communication protocol

The bundles, described in the previous subsection, communicate through DomoML-com messages.

DomoML-com is an XML-based format XML that defines the message structure as shown in Figure 12 .

<fromdevice> light_11</fromdevice>
<service>turning</service>
<serviceStatus> ON </serviceStatus>
<time>….</time>

Figure 12 DomoML-com Message

2.3 Examples of existing environmental control applications

Section 2.3 gives examples of existing environmental control applications, both eye tracking driven and conventional application interfaces. The section then give the critique of these interfaces based on the existing recommendations for accessible environmental control guidelines as given in section 2.1.

2.3.1 Eye-tracking driven

Gaze driven environmental control interfaces can be divided into two methods of control, these are:

- Indirect control – the user may gaze at an interface (such as a computer screen) where representations of the real-world objects are shown and controlled by gaze via the screen
- Direct control – the user may gaze at the actual real-world objects, which then are controlled by gaze directly on the objects

Indirect control interfaces are essentially the same in operation as any other graphical user interface, except that they are specially adapted to be suitable for gaze control. Direct interfaces are an evolution away from indirect interfaces, and attempt to completely remove any computer interface between the user and their environment.

LC Technologies and Erica - As introduced in deliverable 2.4, both LC Technologies and Erica provide basic indirect gaze controlled environmental control programs. These systems provide basic control of lights and appliances anywhere in the home. To use the systems the user turns appliances on and off by looking at a bank of switches displayed on the screen, with commands sent to home sockets and light switches via the home mains electricity wiring via an X10 protocols.

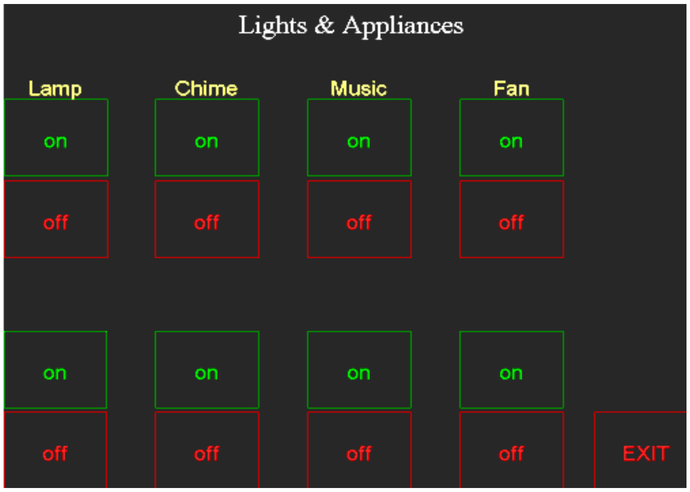

Figure 13. LC Technologies Lights and Appliances program[2]

The LC Technologies system goes some way toward environmental control, but is still quite limited in its functionality, as can be seen from Figure 13. The ERICA system[3] offers essentially the same basic functionality as the LC Technologies system, by presenting the user with a grid of buttons relating to environmental objects, again with a mains-borne X10 communication system. In addition, the user may control an infrared transmitter that can be programmed with TV and other commands to control IR domestic environments.

GazeTalk – GazeTalk is an on-screen application for text entry that is also being adapted for indirect environmental control. Here, the alphanumeric characters currently displayed on gaze talk would be replaced by textual descriptions of the objects within the environment that could be controlled by the user. This is still undergoing development and is the subject of deliverable D4.6 within COGAIN.

Intelligent home interface - this interface for environmental control systems was designed for a head pointing devices which have a greater positional accuracy than gaze devices. This can be seen on the illustration (Figure 14), where the environmental control buttons are quite closely spaced.

2 http://www.eyegaze.com/2Products/Disability/Disabilitymain.htm
3 http://www.eyeresponse.com/Disabilities/Environmental_Control.aspx

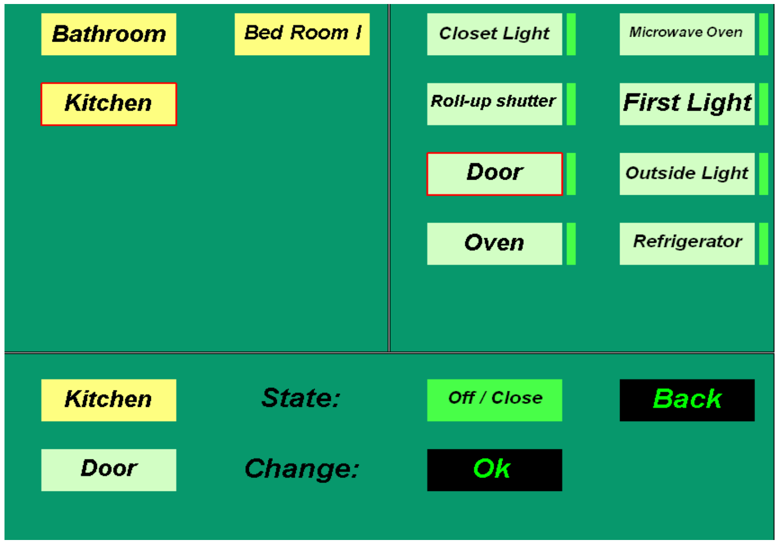

Figure 14. Intelligent home simple interface – high pointing precision

An alternative version of this interfaces currently being developed for gaze-based interaction and on this interface the buttons used to control the environment are well-separated, and due to that separation can react to gaze pointing in their vicinity and do not need gaze to be accurately placed on the buttons (Figure 15).

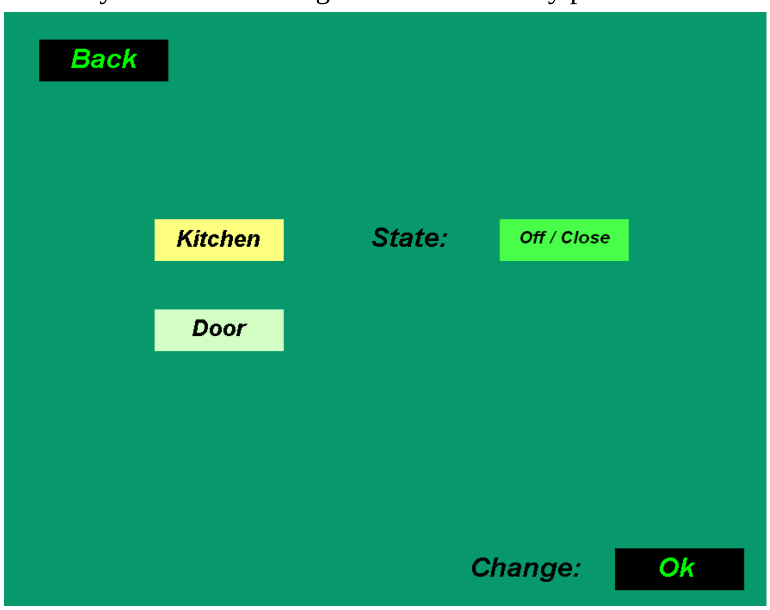

Figure 15. Intelligent home simple interface – low pointing precision

Attention responsive technology interface - as described in the period deliverable the 2.4, the ART (Attention Responsive Technology) interface[4] aims to facilitate direct interaction with the environment of the user by developing a system which responds directly to users' gaze to enable interaction with the environment by

4 'Direct Gaze-Based Environmental Controls' - Alastair G. Gale, Fangmin Shi, Kevin Purdy, COGAIN NoE, to be presented at the 2nd Conference on Communication by Gaze Interaction – COGAIN 2006: Gazing into the Future. Available online at http://www.cogain.org/cogain2006/COGAIN2006_Proceedings.pdf

responding appropriately to the user's 'attention'. In the system 'Attention' is anticipated through means of gaze tracking, with elements of the environment subsequently actuated and controlled by gaze alone via the environmental control system. Figure 16 shows the principle of operation. Here the user looks at a controllable object (1), the interface responds by offering only the appropriate controls (2), and the user operates the object using a custom control interface (3).

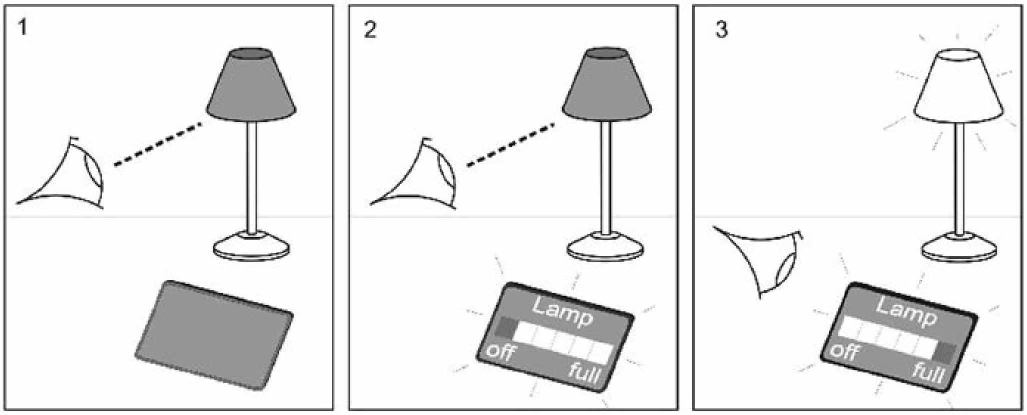

Figure 16. ART system direct interaction

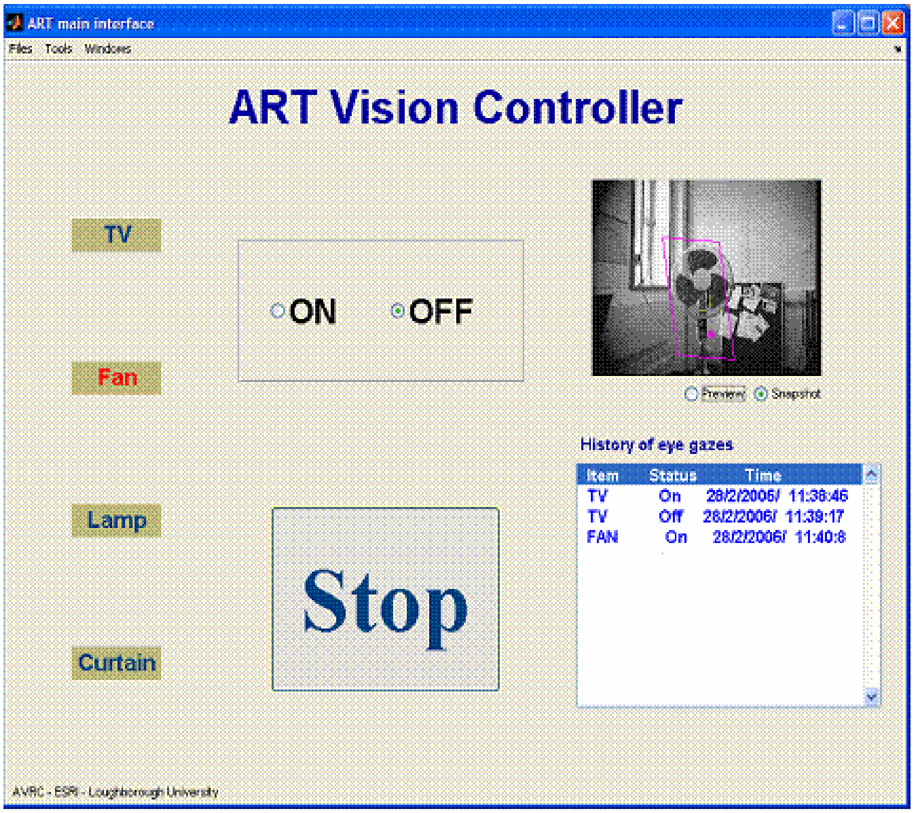

Figure 17. ART indirect controller interface

2.3.2 Non eye-tracking driven

Non eye tracking driven interfaces were introduced in the previous deliverable D2.4 old interfaces in at the call user interface-based systems. These are interfaces that are displayed on a computer screen, and consist of graphical elements representing the various home control functions of the system. Examples of these systems are shown in the next Figures[5].

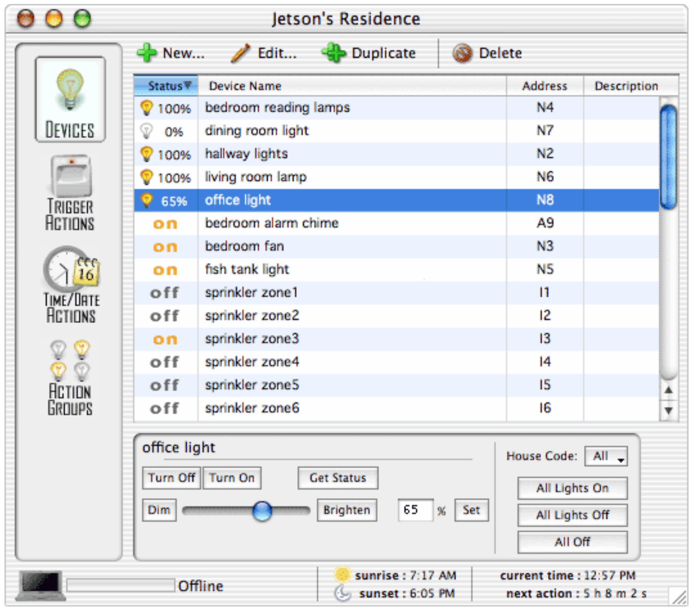

Figure 18. Indigo home control software 1430

Figure 19. PowerLinc V2 home control software

5 www.smarthome.com

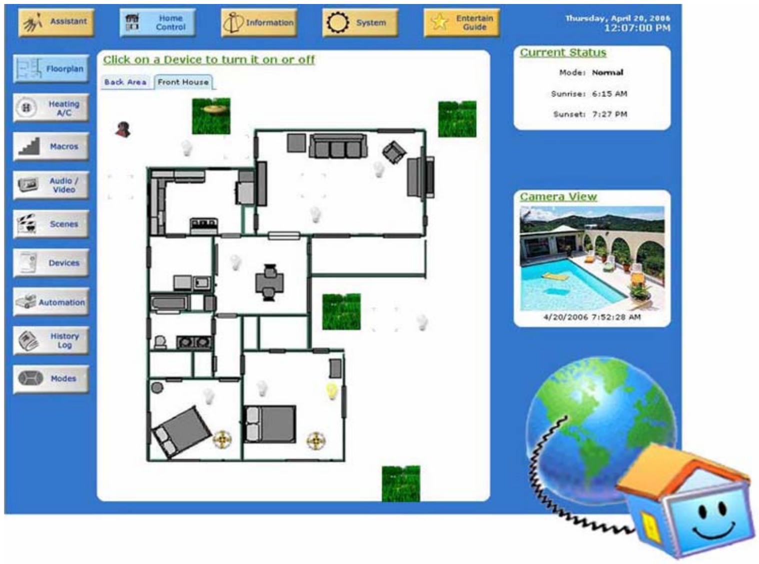

Figure 20. Central Home Automation Director

All of the graphical computer based systems were found to have essentially the same properties of on-screen buttons and menu items to control functions, together with some form of pictorial feedback on the state of the system.

Figure 21 Domotica Labs Direxion interface

Other systems use web-based control application as shown in Figure 21. These kind of applications should respect W3C WAI guidelines to be accessible.

2.4 Critique of existing systems based on existing guidelines review

The operation and suitability for gaze based accessible environmental control of these systems can be assessed using the summary of existing guidelines (Table 1 'Summary of exciting recommendations' from section 2.1.4). The following tables take the existing interface elements of the recommendations and apply them to the existing systems found.

It is notable that none of the existing systems comply with all or even many of the recommendations, clearly showing the need for greater awareness and compliance of the existing recommendations for the proposed COGAIN system, as well as any other systems which may be developed.

Recommendation	Existing recommendation where appropriate to existing interfaces	Surveyed existing system				
		LC Technologies (gaze driven, indirect)	GazeTalk (gaze driven, indirect)	Intelligent Home Interface (gaze driven, indirect)	Attention Responsive Technology (gaze driven, direct)	GUI based interfaces (non gaze driven, indirect)
CEN ELEC TC 205	Create consistent user interface	✓ Consistent	✓ Consistent	✓ Consistent	✓ Consistent	✘ Inconsistent
	Standardize user interface of smart house systems	✘ Not standardised	✘ Not standardised	✘ Not standardised	✘ Not standardised	✘ Not standardised
	Make smart house systems interoperable	✘ Not a feature	✘ Not a feature	✘ Not a feature	✘ Not a feature	✘ Not a feature
	Require upward compatibility among different system versions	✓ Possible	✓ Possible	✓ Possible	✓ Possible	✓ Possible
	Create standard choice of different interfaces, individually adapted	✘ Single choice	✘ Limited choice	✘ Limited choice	✘ Single choice	✓ Possible
	Provide failsafe features and manual systems override	✘ Not a feature	✘ Not a feature	✘ Not a feature	✘ Not a feature	✓ Possible
	Create standards on feedback typologies for all major user groups.	✘ Not done	✘ Not done	✘ Possible but not done at this time	✘ Not done	✘ Not done

Table 2. Compliance with CTC205

Recommendation	Existing recommendation where appropriate to existing interfaces	Surveyed existing system				
		LC Technologies (gaze driven, indirect)	GazeTalk (gaze driven, indirect)	Intelligent Home Interface (gaze driven, indirect)	Attention Responsive Technology (gaze driven, direct)	GUI based interfaces (non gaze driven, indirect)
CEN ELEC TC 122	Create standardised interface to control and input devices across smart house systems and suppliers	✘ Not standardised	✘ Not standardised	✘ Not standardised, but may be implemented	✘ Not standardised	✘ Not standardised
	Convergence of different modes for operation	✘ Single mode	✓ Possible	✓ Possible	✘ Single mode	✓ Possible

Table 3. Compliance with CTC122

Recommendation	Existing recommendation where appropriate to existing interfaces	Surveyed existing system				
		LC Technologies (gaze driven, indirect)	GazeTalk (gaze driven, indirect)	Intelligent Home Interface (gaze driven, indirect)	Attention Responsive Technology (gaze driven, direct)	GUI based interfaces (non gaze driven, indirect)
CEN ELEC other TC's	Standardised symbols e.g. for on-off.	✘ Not standardised	✘ Not standardised	✘ Not standardised, but may be implemented	✘ Not standardised	✘ Not standardised

Table 4. Compliance with CTC other TC's

Recommendation	**Existing recommendation where appropriate to existing interfaces**	**Surveyed existing system**				
		LC Technologies (gaze driven, indirect)	**GazeTalk (gaze driven, indirect)**	**Intelligent Home Interface (gaze driven, indirect)**	**Attention Responsive Technology (gaze driven, direct)**	**GUI based interfaces (non gaze driven, indirect)**
COGAIN D3.3 and D4.1	Resizable cells and grids	✓ Yes	✓ Yes	✓ Yes	✘ No	✘ Not normally available
	A range of input methods	✓ Yes	✓ Yes	✓ Yes	✘ No	✓ Possible
	Choice of symbol vs. text input	✘ No	✓ Possible	✘ No, but could be done	✘ No	✘ No
	A selection of text styles and colours	✓ Yes	✓ Yes	✓ Possible	✘ No, but possible	✘ Not normally available
	A range of speech facilities	✘ No	✘ No	✘ No	✘ No	✓ Possible
	A choice of language	✓ Yes	✓ Yes	✓ Possible	✘ No	✓ Possible
	Easy shift between national language and English version	✘ No	✘ No	✘ No	✘ No	✘ No, but possible via operating system
	Adjustable speed	✓ Yes	✓ Yes	✓ Yes	✘ No	✘ No
	Access to system commands from within the eye-writing system	✓ Possible	✓ Possible	✓ Possible	✘ No	✘ No

Table 5. Compliance with COGAIN D4.1

3 Problem analysis

3.1 Information exchanged with an Environmental Control system

Two different phases can be identified when Control Applications and Environmental Control Systems exchange information:

- A **Start-up phase**: start-up and configuration data are exchanged only when the control applications starts and they are used to configure the connection between Control Applications and the Environmental Control systems, as well as to build the control applications user interface.
- An **Operational phase**: in this phase, operative data such as commands and events are the main content of the communication.

Figures 22 and 23 show some particular examples of information that could be exchanged between a Control Application and an Environmental Control System during the start-up and operational phases, respectively. The following subsections briefly describe each blocks of the figures. In particular, the information that is generally present in every environmental control system will be underlined, while the other are usually optional or system dependent.

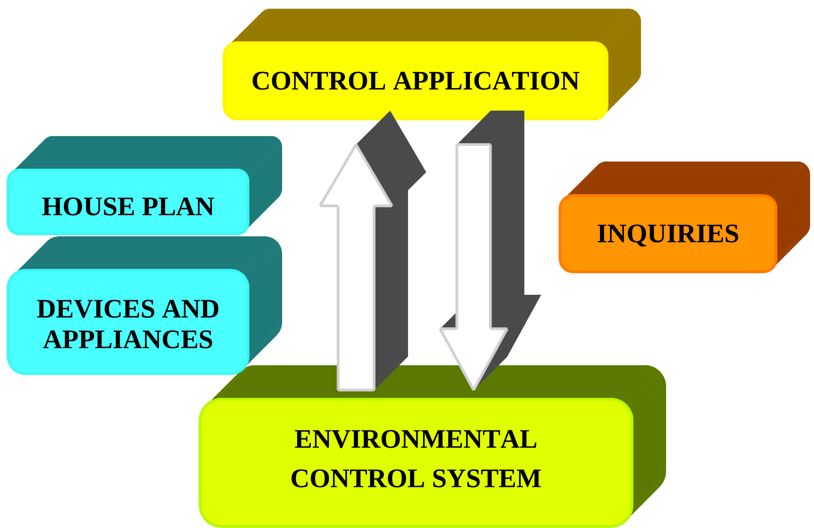

Figure 22. Information exchanged with a Domotic System during the start-up phase.

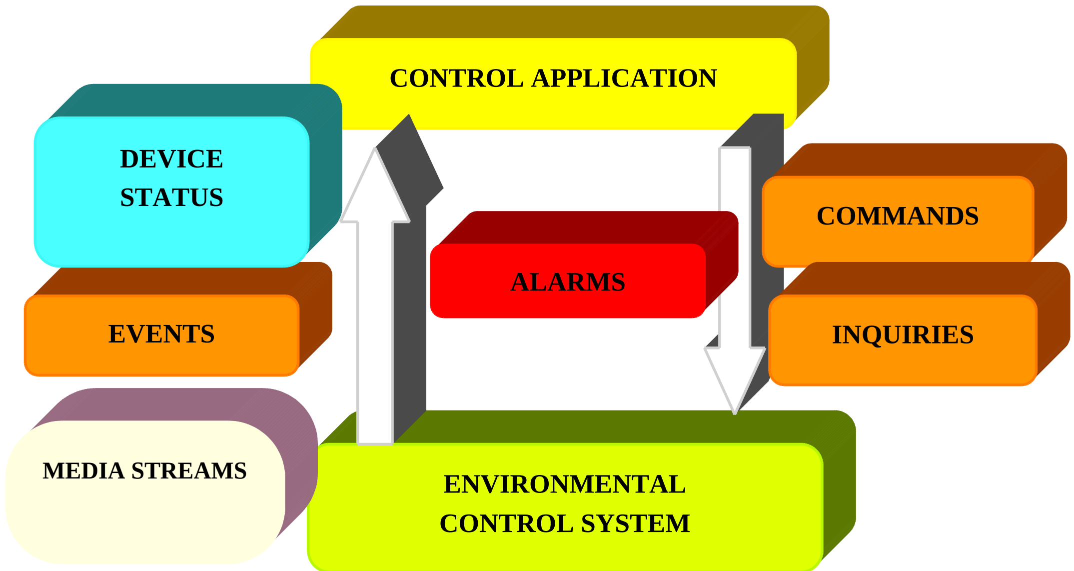

Figure 23. Information exchanged with a Domotic System during the operational phase.

3.1.1 House PLAN

Typically, control applications need to know the house plan with more or less details. These data are necessary to build the application user interface.

Example of house plan information could be:

- Room type: it is the purpose of the room, e.g. living room, bedroom, bathroom, …
- Room description: is a short description that could appear on the user interface, e.g. Ron's bedroom, bathroom without shower, etc.
- Room location: represents the room position in relation with the other rooms. This information could be stored in different ways:
  - Text notation: used only as further description, e.g. on the left of the kitchen, between bedroom and living room, etc.
  - Geometric notation: used to show a map of the controlled environment, e.g., coordinates of centre and dimensions for a rectangular room plan or coordinates of set of points for more complex room plans.

3.1.2 Devices and Appliances

Control applications also need to know which kind of devices are presents in the rooms of the house. These data are necessary to build the application user interface and to allow the actual control of the home devices.

Example of device information could be:

- Type: identifies a list of capabilities that a device owns. It is possible to create a hierarchical tree of types in order to allow inheritance, as shown in Figure 24[6], between Fridge and Deep freezer, the latter offering additional functionalities (lower temperature, faster freezing, etc.) with respect to those inherited from the former. The type definition is also necessary for reasoning capabilities of

[6] This figure is taken from **DomoML-env: an ontology for Human Home Interaction** (Lorenzo Sommaruga , Antonio Perri, and Francesco Furfari)

the system; for example, the command “lower the temperature of all fridges” can’t work correctly if the system doesn’t know devices hierarchy.

- Description: it is a short description that could appear on the user interface to identify the device.
- Manufacturer: it is the name of the device producer. This information together with Type is essential in multi-brand home automation systems to select the appropriate device drivers.
- State: indicates which and how many states a device owns. Figure 25 shows example of 2-state and 5-state devices.
- Location: represents the room of the house where the device or appliance is located.
- Address: it indicates the way to access the devices. It can be different for each home automation system; some example of addressing are:
    - IP address: generally used in Ethernet based home system, e.g. *172.0.0.3* for the Fridge
    - Location address: the address is composed by room and device names, e.g. *home.kitchen.fridge* for the Fridge.
    - ID address: the device has an unique identifier that can be a number, a name or a code, e.g. *0x1234* for the Fridge

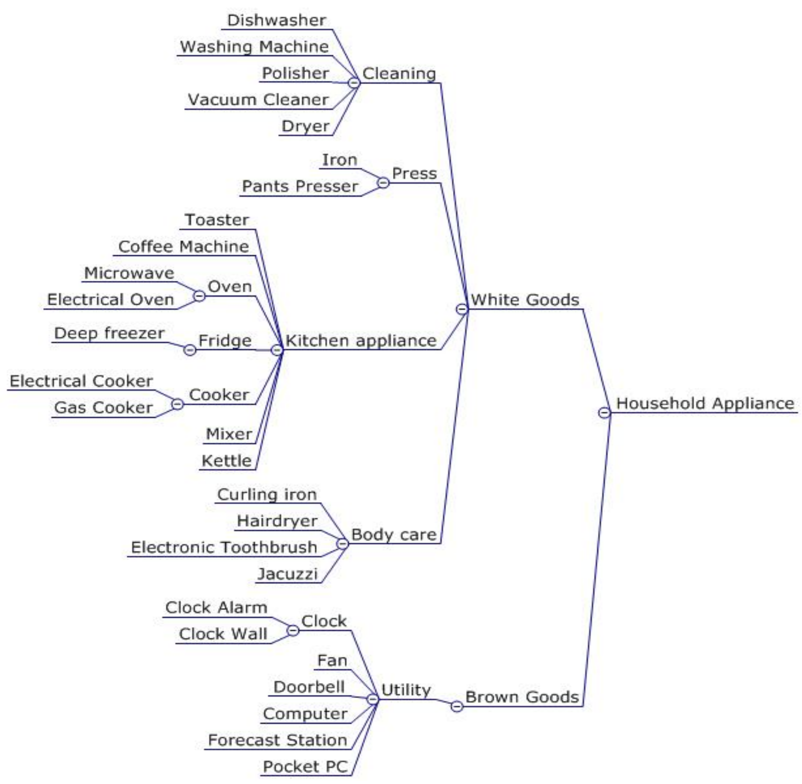

Figure 24. Tree of device types

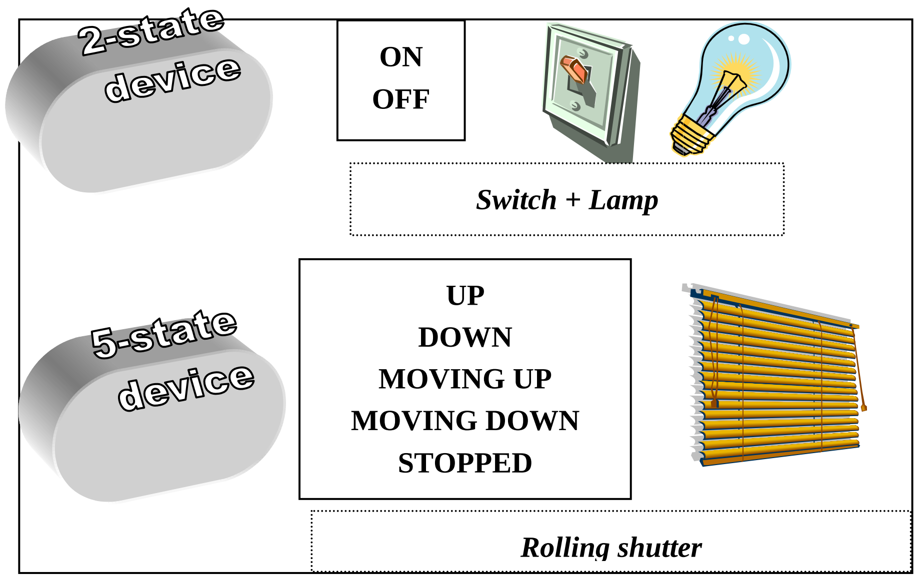

Figure 25. Examples of the state information

3.1.3 Inquiries

Inquiries are information requests from a control application toward an environmental control system. These messages are sent either in the start-up or in the working phase. Examples of inquiries are:

- Request of House Plan information.
- Request of the devices information.
- Request of a Device Status.
- Registration to listen an event.

3.1.4 Device Status

This message is sent from an environmental control system to a control application as response to an inquiry (i.e., the request of device status). The data contains the information requested, e.g., the fridge temperature, whether the door is closed, the shutter position (up, down, rolling), etc..

3.1.5 Events

Events are messages sent from the environment control system to the control application to signal a change of state of one or more devices, appliances or sensors. Unfortunately not all home automation system are event driven, so the control application sometimes has to emulate events by polling the state of devices.

The event-message can have several formats. In the system described in subsection 2.2.3, for example, the format is DomoML-com.

Mainly there are two modalities to manage events:

- Registered: the control application requests and registers the events that it need to listen to home automation system. The home automation system sends to the control application only the registered events.
- Sniffing: the control application receives all events from home automation system and uses only what it needs.

3.1.6 Alarms

Alarms are events with maximum priority. These events can be sent either from home systems or from control applications. Examples of alarms sent from home system are:

- Intrusion detection.
- Fire / gas alarms.
- Device or appliance malfunction.

Examples of alarms sent from the users through control application are:

- STOP: at any time, the user can stop what the home system is doing.
- Medical-Request: the home system could be programmed to request assistance.

Control applications have to handle alarms quickly and safely. Further examples of alarms management are reported in the next chapter.

3.1.7 Multimedia Streams

Some devices such as webcam, tv or stereo can produce multimedia streams. Typically these streams must be treated in different way compared with events and commands because they are real-time data and need a wide bandwidth. There are several protocols that can be used to transmit streams on a network, one of the most used being the Real Time Protocol (RTP).

3.1.8 Commands

Commands are messages sent from control application to home system. Examples of commands are:

- Request of house configuration
- Request of device or appliance state
- Request of multimedia stream
- Activate/Deactivate an appliance
- Change the state of a device
- Set house system configuration

Commands, like events, can have different priorities.

3.2 Example Use Cases

In the next subsections several hypothetic situations are described, in which disabled users interact with a domotic system using a gaze-based control application.

The described systems are ideal systems; they work how we would want them to work. Keynotes after each use case show which element(s) of the scenario/functionality are desirable in a control application.

The following examples concern different gaze based users, from living alone with part-time assistance, to fully dependent.

3.2.1 Turning off the Bathroom Lights

The user was playing chess on the pc while his sister was cleaning the bathroom.

After two hours, he finally won the match and decided to watch a movie while his sister went out with a friend. While he was watching the movie, he realizes that his sister has left the bathroom lights on, by glancing at the yellow colour of the background of the icon corresponding to the same lights or by looking at the message on the bottom of the screen. Therefore, he turns off the lights, acting on control application interface. The lights icon starts shading to grey and the status bar text shows “bathroom light is turning off”. He sees that the command has been executed by noticing the grey colour of the background of the lights icons, and then he returns at the normal visualization, still looking at the movie.

Keynotes:

- *The control application shows the device status possibly in redundant way (i.e., icons and status bar text).*
- *A visual feedback is given to the user while a command is executing and upon its conclusion (icon color and status bar).*

3.2.2 Intrusion Alarm and Room Vision

The user is watching the television news. He feels safe: he has activated the antitheft alarm in all the other rooms. Suddenly the television set switches from the normal vision to the alert modality for the domotic house, by visualizing the menu of the bedroom and the message of a possible intrusion in that room. The domotic system, thanks to a camera installed in the bedroom, and through a motion detection algorithm, identifies some suspicious movement in the room. The television set now shows the image captured by the bedroom camera. The user sees that his lovely cat, Garfield, just woke up after its daily siesta on the bed, and it is now walking towards the kitchen, to have its dose of biscuits[7] and hugs. The user, after a sudden fright, is now relaxed and feels even safer.

Keynotes:

- *An alarm is visible and clear, each other activity that could interfere with the alarm notification and visualization is temporarily suspended.*
- *A default action is executed after alarm notification (in this case a camera stream is shown)*

3.2.3 Fire Alarm

The user is surfing the internet, while in the kitchen a failure of the electrical plant starts a fire. Immediately the control application shows the images of the kitchen, plays loud sounds and proposes two clear options ANTI-FIRE or FALSE ALARM. The user is agitated, this time it is not her feebleminded husband that smokes a cigarette in the kitchen, it’s a real fire, so she loses control of the gaze tracking system. After a few seconds, since the user doesn’t select any of the two options, for safety the anti-fire procedure is automatically

[7] the real Garfield would have preferred lasagna, of course

selected (as if the user had selected the ANTI-FIRE option): all windows and doors are closed, a shower of foam quickly extinguishes the fire and the emergency numbers are called.

Keynotes:

- *The control application immediately shows the alarm and plays sound to capture attention*
- *The control application shows few clear options to manage the alarm.*
- *After a short time-out the system automatically selects the safest option.*

3.2.4 Not only eyes

Unfortunately, the infrared camera of the gaze tracker is broken and two or three days are necessary to repair it. The user can still use the control application with a switch or with her old and less accurate eye tracker that lies in the cellar covered with dust. The user feels comfortable to use a control application with the same user interface, even if with a different input device.

Keynotes:

- *The control application supports different input devices and different interaction methods (selection mode, gaze, mouse)*
- *The interface of the control application is only slightly changed, so the user does not have to learn again how it works.*

3.2.5 Turn off the lights for the night

It is nightfall and the user wants to sleep. He simple selects the “Go to sleep scenario” created previously. The control application interface shows all the commands in progress: turn off the lights in the corridor, lock the door, close the window of the bedroom, etc. When the window is closing, the user decides that this evening he would rather feel the spring gentle breeze, so he stops the command. The control application allows the user to decide whether to stop only the current command or the whole scenario.

Keynotes:

- *The control application can manage (create, modify and delete) scenarios, that are lists of commands.*
- *The scenario execution progress is notified to the user, who at any time can stop it.*
- *The control application gives priority to user commands instead of automated actions.*

3.2.6 Losing eye control

The user is using an eye tracker system very sensible to head movements. Unfortunately, he involuntarily tilts his head and loses the control of eye tracker. The control application receives incongruent commands and “understands” that the eye control is lost by analyzing eye movement patterns. The application interacts with the eye tracker driver and re-starts a calibration session. After some failed attempts, the control application sends an alarm event to the house control system that plays a sound and flashes lights to capture attention. The user’s brother, who is cooking in the kitchen, can reseat the user in the right position.

Keynotes:

- *The control application works in strong relation with eye tracker, it can access to eye movement data, start a new calibration session, etc.*
- *The control application can request human assistance through the home control system.*
- *The user has the possibility to arrange the problems by himself before requesting help.*

3.2.7 Multimodal and alternative inputs

The user wears a head-mounted eye tracker and moves around the house on his wheel chair.

He looks at the window and says «close», but the control application misunderstands what he's looking at and sends the message "close the door" to the home automation system. The user, immediately, says «STOP» and the door is re-opened. Then, the user moves closer to the window, gazes at it and says again «close». The control application now does the correct work.

Keynotes:

- *The control application can have no visual interface.*
- *The control application supports multimodal inputs (gaze, speech).*
- *The user can interrupt the execution of any command.*

3.2.8 The telephone rings

The user is reading the last Harry Potter novel on his PC, while listening his preferred music.

When he is immersed in those pleasant activities, the telephone rings. The control application slightly lowers the music, shows the caller number and offers two options: ANSWER or DROP. The user doesn't now the number and selects ANSWER. The control application stops the music and activates the speakerphone. After a while the phone call finishes and the control applications restarts the music. It was just a *muggle* that called the wrong number.

Keynotes:

- *The control application executes automatic actions driven by events (e.g., a phone call).*
- *The control application shows clear options.*
- *The control application can restore the previous conditions when a particular operation is finished.*

3.2.9 Not a funny joke

The user is a playful person and asks to her friend to take a book in her bedroom on the second floor. The user follows her friend's movements through the control application that shows the images captured from the several cameras in her house. When her friend arrives in the bedroom, she wants to make a joke. She quickly sends a list of commands: lock the door, play loud music, close the window, open the windows, turn the lights on, turn the lights off, etc.

She is enjoining while her friend appears scared, but, after a while, her friend pushes a switch on the wall in the bedroom and all those strange things end.

Keynotes:

- *The control application can quickly execute list of commands.*
- *The control application gives priority to the commands coming from the home instead of software ones.*

4 Application requirements

4.1 General philosophy and structure

These guidelines explain how to make control applications, for home automation system, accessible. The guidelines are intended for all eye tracking applications developers. The primary goal of these guidelines is to promote safety and accessibility. The structure of this section is strongly inspired by W3C Web Content Accessibility Guidelines.

The guidelines are generated from all of the previous sections, in particular from the user cases.

4.2 Overall Design Guidelines

Each guideline has a priority level based on the impact on safety and accessibility:

- **Priority ❶** - A smart house application developer **must** satisfy these guidelines.
- **Priority ❷** - A smart house application developer **should** satisfy this guideline.

4.2.1 Category 1: Control applications safety

Guideline 1.1 *Provide a fast, easy to understand and multimodal alarm notification.* ❶

A user needs to notice as soon as possible that the environmental control system has sent an alarm. The control application should notify the alarms in several ways, e.g., with sounds, flashing icons, text messages.

Guideline 1.2 *Provide the user only a few clear options to handle alarm events.* ❷

Several gaze trackers are less accurate when the users are agitated, therefore in case of alarm the control application should propose only a limited but clear set of options (3 at most).

Guideline 1.3 *Provide a default safety action to overcome an alarm event when the user does not decide.* ❶

In case of emergency the user could lose the control of the input device, therefore the control application should take the safest decision after a time-out. The time-out length should be dependent on the alarm type.

Guideline 1.4 *Provide a confirmation request for critical and possibly dangerous operation.* ❶

With inaccurate or badly configured Gaze trackers the *Mida's touch error* can be frequent, i.e., each object/ command gazed from the user is selected/executed, therefore the control application should request a confirmation for possibly dangerous operations.

Guideline 1.5 *Provide a STOP functionality that interrupts any operation.* ❶

In some occasions, the environmental control system can operate actions that the user does not want, e.g., a selection of a wrong command, or automated and prescheduled scenarios, or the user changes idea, etc. The control application should allow a STOP method for interrupting any operation.

4.2.2 Category 2: Input methods for control application

Guideline 2.1 *Provide a connection with the Cogain ETU-driver.* ❶

The Cogain ETU-driver described in deliverable D2.3 is a single gaze communication standard that allows any third party application to be driven by a range of different eye tracking hardware systems. By using the driver, there is no need for any third party application to be changed or recompiled when switching between differing eye tracking hardware systems.

Guideline 2.2 Support several input methods. ❷

The gaze tracker, unfortunately, can break down, therefore the control application should support also alternative input methods, e.g. switch (scansion mode selection), keyboards, mice, etc.

Guideline 2.3 Provide reconfigurable layouts, appropriate for different eye tracking performances and user capabilities. ❷

Eye trackers have a very wide performance range; therefore, a control application should have a reconfigurable visual interface adaptable to different resolutions and precisions of the eye trackers.

Guideline 2.4 Support more input methods at the same time (multimodal interaction). ❷

The user could be able to use alternative input channels beyond the gaze, e.g. voice, fingers movements, etc. The control application should support the combination of more input method at the same time, for example selection with gaze and click with mouse.

Guideline 2.5 Manage the loss of input control providing automated default actions. ❷

The control application should "understand" when the user has lost the eye tracker control and should provide default actions (e.g. recalibration, play an alarm, etc.).

4.2.3 Category 3: Control applications operative features

Guideline 3.1 *Respond to environmental control events and commands at the right time.* ❶

The control application should be responsive: it should manage events and commands in an acceptable time slot.

Guideline 3.2 *Manage events with different time critical priority.* ❶

The control application should distinguish between events with different priority. The time critical events must be acted upon with a short fixed period (e.g. fire alarm, intrusion detection).

Guideline 3.3 *Execute commands with different priority.* ❶

The home automation systems commonly receive more commands at the same time (e.g. different users, scenarios, …). The control application should discriminate commands with different priority and should adopt a prefixed management policy.

Guideline 3.4 *Provide feedback when automated operations or commands are executing.* ❷

Scenarios, selected by the user, could include several scheduled commands. The control application should show the actions in progress and inform the user when a scenario is terminated.

Guideline 3.5 *Manage (create, modify, delete) scenarios.* ❷

Repeating a long sequence of commands to do a frequent task could be tedious for the user. It is necessary for gathering list of commands and manage them as a single one. The control application should allow creation, modification and deletion of scenarios.

Guideline 3.6 *Know the current status of any devices and appliances.* ❷

The control application should know the current status of any devices and appliances of the home, in order to show that information and to take "smart" automated decision (e.g. prevent a dangerous condition, activated energy saving plan, etc.).

4.2.4 Category 4: Control applications usability

Guideline 4.1 *Provide a clear visualization of what is happening in the house.* ❶

In accordance with the guidelines of category 1 and 3, the control application interface should provide a clear, easy understandable visualization of the execution progress of the commands.

Guideline 4.2 *Provide a graceful and intelligible user interface.*❷

Consistent page layout, easy to understand language, and recognizable graphics benefit all users. The control application should provide a graceful and intelligible user interface, possibly using both images and clear texts.

Guideline 4.3 *Provide a visualization of status and location of the house devices.* ❷

The control application should show the house map containing, for each room, a representation of the devices and their status.

Guideline 4.4 *Use colours, icons and text to highlight a change of status.* ❷

The control application interface should highlight the device status change using images, texts and sounds.

Guideline 4.5 *Provide an easy-to-learn selection method.* ❷

In spite of the control application could present complex features and functionally, it should provide an usable, easy-to-learn interaction method.

4.3 Next steps - Implementation and simulation

Politecnico di Torino will release a domotic house simulator, compliant with Cogain recommendations, in the beginning of 2008. Cogain partners should develop several control application based on their existing works (GazeTalk, Saw, Dasher, etc..).

5 Conclusions

This deliverable set out to define draft standards for gaze-based environmental control. The deliverable sets out the need for these systems to be both enabling and also reducing the workload of the users of these systems. It complied a set of tables reviewing and examining existing systems and interfaces in the context of existing guidelines and shows that no current systems achieve close compliance with these guidelines for accessibility and control. A new proposed COGAIN architecture for domotic systems is proposed that will allow systems to be constructed that comply with the underlying operational guidelines, and goes on to give a problem analysis that highlights the interface and underlying operational requirements for gaze-driven systems. Finally design guidelines with operational priorities are given that highlight the safety issues and ease-of-use issues for gaze control.

The deliverable shows that although dedicated gaze-driven systems are available, these do not follow all of the existing guidelines, and that no existing systems are particularly suitable for gaze control when that control is used for flexible and interoperable systems. It presents a new architecture and a new set of design guidelines that will now be followed and implemented within COGAIN to give a fully compliant gaze-driven domotic system for users.