

# Eye-based Direct Interaction for Environmental Control in Heterogeneous Smart Environments

Fulvio Corno, Alastair Gale, Päivi Majaranta, and Kari-Jouko Rähä

## 1 Introduction

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, 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 to be 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:

---

Fulvio Corno

Politecnico di Torino, Dip. di Automatica e Informatica, Corso Duca degli Abruzzi 24, 10129 Torino, Italy, e-mail: [fulvio.corno@cogain.org](mailto:fulvio.corno@cogain.org)

Alastair Gale

Applied Vision Research Centre, Loughborough University, Garendon Wing, Holywell Park, Loughborough LE11 3TU, UK, e-mail: [alastair.gale@cogain.org](mailto:alastair.gale@cogain.org)

Päivi Majaranta

Unit for Computer-Human Interaction (TAUCHI), Department of Computer Sciences, FI-33014 University of Tampere, Finland, e-mail: [paivi.majaranta@cogain.org](mailto:paivi.majaranta@cogain.org)

Kari-Jouko Rähä

Unit for Computer-Human Interaction (TAUCHI), Department of Computer Sciences, FI-33014 University of Tampere, Finland, e-mail: [kari-jouko.raihä@cogain.org](mailto:kari-jouko.raihä@cogain.org)

Reduce the day to day home operation workload of the occupant

There are several terms that 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 chapter environmental control within the home as a form of assistive technology to aid users with disabilities 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—and within this chapter—“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 the user may not be able to accomplish.

Users with a physical disability 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 such a person can control their environment. Hence, extending the definition above, 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 people as they can control the World around them.

Typical targets are those items within the home that may be easily controlled, and which the user would often wish to control. Examples include:

- Lighting (switches, dimmers);
- Windows and blinds;
- Door lock;
- Intercom (microphone and button at the door as well as bell and loudspeaker in the apartment);
- Heating, Ventilating, and Air-Conditioning (HVAC);
- Home appliances (also called white goods—domestic equipment such as refrigerators, cookers, washing machines and central heating boilers);

- Brown goods (name for Audio Visual Appliances and devices);
- Home security devices (e.g., burglar alarm, fire alarm);
- Telecommunication;
- Information access.

However, such a system is limited as it requires the user to actively and deliberately control each device. This may become tiresome and cause fatigue over time. Many functions could be automated to save the user the efforts of controlling the system themselves. Thus, an improvement to direct control is to add a level of “intelligence” to the system.

Electronic devices that contain processors or are computers that can communicate with other systems are increasingly popular in ordinary homes. In order to address security and control, communications, leisure and comfort, environmental integration and accessibility, it is desirable to have a central control system to integrate and manage all the devices throughout a domestic home. Therefore, it is essential that this environment control system can interact and work together with all the electronic devices within the home to benefit the users in the home.

Most modern homes have appliances that allow some degree of remote control. Environmental control aims to integrate and extend this control throughout the home. A home with an environmental control system installed might have many computers, perhaps built into the walls, to allow the homeowner to control applications in any part of their home from any other, or it may have a single simple computer operating an essential function such as emergency call and security. In addition to integration, intelligent systems are being developed. The great potential of these systems lies in their ability to apply computer programming to act in what could be called “intelligent” ways.

## **2 The COGAIN network**

COGAIN (Communication by gaze interaction) is a Network of Excellence funded by the European Commission in 2004–2009. COGAIN has brought together researchers, users, user organisations, and eye tracking manufacturers to foster the development of solutions for users with severe motor disorders. Eventually the only way for communication for such users may be the use of their eyes, the modality that COGAIN focuses on. This can be caused by a number of diseases, including Amyotrophic Lateral Sclerosis; Motor Neurone Disease; Multiple Sclerosis; Cerebral Palsy; Spinal Cord Injury; Spinal Muscular Atrophy; Rett Syndrome; Stroke and Traumatic Brain Injury. The potential numbers of such users in the EU alone has been estimated as circa three million. In addition other users, who have some minor motor or mobility impairment, could also use an eye based system as an optional interface mechanism to augment existing controls.

COGAIN was launched with several goals in mind. The different stakeholders involved did not previously have a forum for meeting and working together, and the network has provided unprecedented opportunities for them all to share their

expertise. When COGAIN started, the application base available for the target user community was rather limited; a second goal of COGAIN was to radically increase both the number of applications and also the application areas where eyegaze could be used. Third, the equipment needed for making use of eyegaze in the interaction has been rather expensive, often prohibitively so for most people who could benefit from the use of eyetrackers. Here COGAIN has worked in two ways: first, researchers have actively sought for solutions that would enable the development of low-cost trackers; and second, COGAIN has tried to influence decision makers, conveying the information that eye tracking is mature, it works, and is both a necessity and a critical life improvement technology for certain users.

As the name of the network indicates, COGAIN started by focusing on communication in the traditional sense: that of producing messages to be delivered to other people, usually in written form. This area, eye typing or gaze typing, is the most established—actively studied and actively used—in the eye tracking field. COGAIN has collected this knowledge together and published recommendations and observations on how to make the installations work for individual users ([10]).

As the work of COGAIN evolved, its scope has expanded from utility applications on the desktop to other types of applications, such as games and entertainment, to supporting mobility of users (for instance for users in wheel chairs), and to environmental control—the focus of this chapter.

The selection of desktop applications that exist today is fairly extensive. A major hindrance preventing full access to these solutions by all users is that they are often proprietary, working on specific platforms. COGAIN has worked to overcome this problem in the desktop environment by proposing standards to enhance interoperability [1], but the take-up of the proposal has been slow outside the network. In environmental control the problems are an order of magnitude more challenging, because several technologies are involved, and there may be alternative interaction and control mechanisms in simultaneous use for them. The next section presents the approach COGAIN has taken to facilitate solutions that are flexible in the presence of multiple solution providers.

Similarly, just as there are many alternative techniques for gaze-based interaction in the desktop environment (for instance, selection by dwell time, selection by eye gestures, or selection by switches), there are two fundamentally different approaches to environmental control by eye gaze: control of an interface on the computer screen, or direct control of the environment by embedding intelligence in the objects in the environment. The second main section of this chapter discusses these approaches, both of which have been investigated in COGAIN, in more detail.

### **3 Domotic Technology**

Domotic systems, also known as home automation systems, have been available on the market for several years, however only in the last few years they started to spread also over residential buildings, thanks to the increasing availability of low

cost devices and driven by new emerging needs on house comfort, energy saving, security, communication and multimedia services.

### ***3.1 Overview of Domotic Technologies***

For the purposes of the current discussion, environmental control systems may be thought of as comprising of five elements:

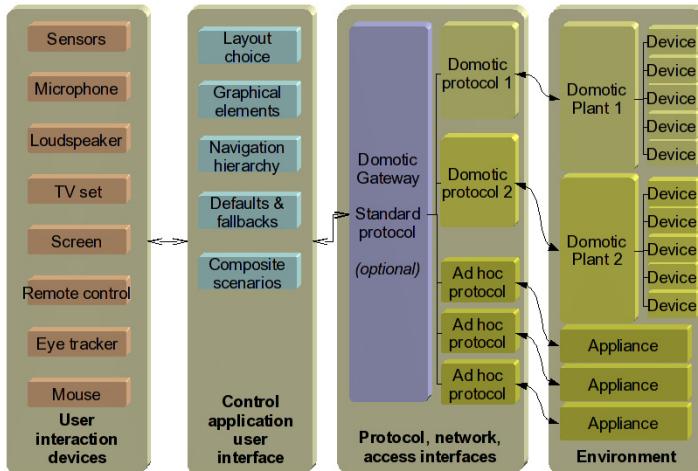
- **The communication capabilities of the user** – the physical capabilities of the user to convey their needs to the environmental control system (such as the ability to click a switch, or point with their eyes).
- **The user interface to the system** – to allow the user to control the environment via the interface by showing the state of the environment, and receiving user control input to that environment (such as a computer screen showing a graphical representation of the environmental state).
- **The central domotic system** – that processes user commands and environmental states and adds integration and intelligence to the system, receiving input from the user and the environment and sending out environmental control signals (the central computer based ‘hub’ of the system).
- **The communications system** – that flows to and from the central domotic system to the environmental actuators and sensors in the domestic environment (the wires or wireless transmission system that links the distributed actuators and sensors in the environment).
- **The domestic environment** – the real-world devices and functions controlled by the system (such as doors, heating systems, curtains, alarms).

A Smart Home for disabled people may include assistive devices (electro-mechanical/robotic systems for movement assistance), devices for health monitoring and special user interfaces. In the case of COGAIN it is a gaze based interface.

From the technical point of view, 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 1. The four main layers, 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.

#### **3.1.1 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



**Fig. 1** Generic Layers of a Domotic System

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, and 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 [2]. It is normally safe to 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 form 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 difficult to classify in a uniform manner and to integrate seamlessly into control interfaces.

**Smart appliances.** Because of 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” [9]. 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.<sup>1</sup> Such interoperability, 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.

### 3.1.2 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 knowledge of 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 they support any standard protocol, or reverse-engineering any proprietary communication features they may pos-

---

<sup>1</sup> DLNA - <http://www.dlna.org/>



sess. 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 have 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, and the varying level of detail and functionality offered by different protocols on different topics.

### 3.1.3 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 the remainder of this chapter:

- *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 and menu labels 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 they have limited time to take important decisions, which may pose threats to their 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 to organize 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 menu 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.

### 3.1.4 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 accommodate 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.

## 3.2 Architecture for Intelligent Domotic Environment

Current domotic solutions suffer from two main drawbacks: they are produced and distributed by various electric component manufacturers, each having different functional goals and marketing policies; and they are mainly designed as an evolution of traditional electric components (such as switches and relays), thus being unable to natively provide intelligence beyond simple automation scenarios. The first drawback causes interoperability problems that prevent different domotic plants or components to interact with each other, unless specific gateways or adapters are used. While this was acceptable in the first evolution phase, where installations were few and isolated, now it becomes a matter of concern as many large buildings such as hospitals, hotels and universities are mixing different domotic components, possibly realized with different technologies, and need to coordinate them as a single sys-

tem. On the other hand, the roots of domotic systems in simple electric automation prevent satisfying the current requirements of home inhabitants, who are becoming more and more accustomed to technology and require more complex interaction possibilities.

In the literature, solutions to these issues usually propose *smart homes* [8, 7, 20], i.e., homes pervaded by sensors and actuators and equipped with dedicated hardware and software tools that implement intelligent behaviors. Smart homes have been actively researched since the late 90's, pursuing a *revolutionary* approach to the home concept, from the design phase to the final deployment. The costs involved are very high and have prevented, until now, a real diffusion of such systems that still retain an experimental and futuristic connotation.

The approach proposed in this chapter lies somewhat outside the smart home concept, and is based on extending current domotic systems by adding hardware devices and software agents for supporting **interoperation** and **intelligence**. Our solution takes an *evolutionary* approach, in which commercial domotic systems are extended with a low cost device (embedded PC) allowing interoperation and supporting more sophisticated automation scenarios. In this case, the domotic system in the home evolves into a more powerful integrated system that we call a Intelligent Domotic Environment (IDE). IDEs promise to achieve intelligent behaviors comparable to smart homes, at a fraction of the cost, by reusing and exploiting available technology, and by providing solutions that may be deployed even today.

Most solutions rely on a hardware component called residential [6] or home gateway [15] originally conceived for providing Internet connectivity to smart appliances available in a given home. This component, in our approach, is evolved into an interoperation system, called DOG (Domotic OSGi Gateway), where connectivity and computational capabilities are exploited to bridge, integrate and coordinate different domotic networks that will be able to learn user habits, to provide automatic and proactive security and to implement comfort and energy saving policies by using low cost, commercially available technologies.

An Intelligent Domotic Environment (IDE, Figure 2) is usually composed of one<sup>2</sup>, or more, domotic systems, by a variable set of (smart) home appliances, and by a Home Gateway that supports implementation of interoperation policies and provides intelligent behaviors.

Domotic systems usually include domotic devices such as plugs, lights, doors and shutter actuators, etc., and a so-called network-level gateway that allows to tunnel low-level protocol messages over more versatile, application independent, interconnection technologies, e.g., Ethernet. These gateways are not suitable for implementing features needed by IDEs as they have reduced computational power and they are usually closed, i.e., they cannot be programmed to provide more than factory default functionalities. However, they play a significant role in an IDE architecture as they offer an easy to exploit access point to domotic systems.

The Home Gateway is the key component for achieving interoperation and intelligence in IDEs; it is designed to respond to different requirements, ranging from

---

<sup>2</sup> in this case interoperation may not be needed but intelligence still needs to be supported

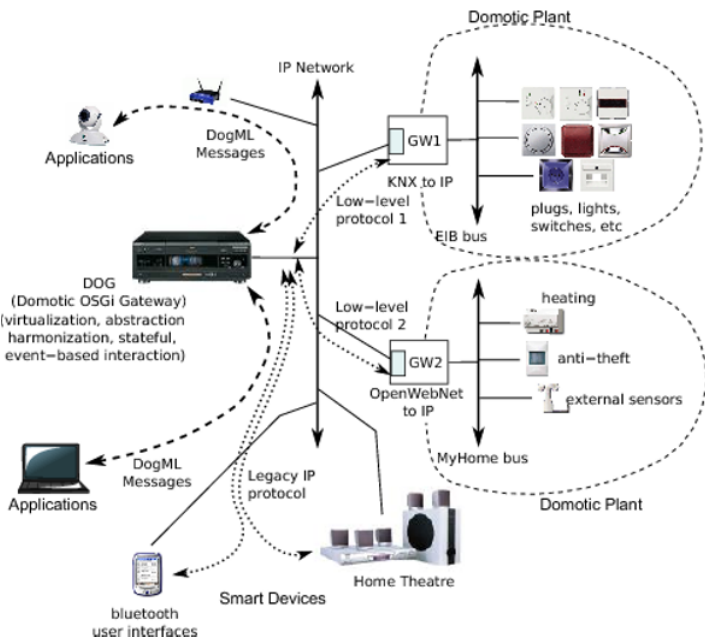


Fig. 2 Intelligent Domotic Environment.

simple bridging of network-specific protocols to complex interaction support. These requirements have been grouped in three priority levels (Table 1): priority 1 requirements include all the features needed to control different domotic systems using a single, high-level, communication protocol and a single access point, priority 2 requirements define all the functionalities needed for defining inter-network automation scenarios and to allow inter-network control, e.g., to enable a Konnex switch to control an OpenWebNet light, and priority 3 requirements are related to intelligent behaviors, to user modeling and to adaptation.

A domotic home equipped with a home gateway is said to be an Intelligent Domotic Environment if the gateway satisfies at least priority 1 and priority 2 requirements. Priority 3 requirements can be considered advanced functionalities and may impose tighter constraints on the gateway, both concerning the software architecture and the computational power.

While priority 1 requirements basically deal with architectural and protocol issues, requirements listed in priorities 2 and 3 imply the adoption of sophisticated modeling techniques. In fact, the cornerstone of intelligent behaviors and applications is a suitable house modeling methodology and language (R2.1). In DOG, we chose to adopt a semantic approach and to adopt technologies developed in the Semantic Web community, to solve this knowledge representation problem and enable DOG to host or support intelligent applications.

**Table 1** Requirements for Home Gateways in IDEs.

Priority	Requirement	Description
R1 Interoperability	R1.1 Domotic network connection	Interconnection of several domotic networks.
	R1.2 Basic interoperability	Translation / forwarding of messages across different networks.
	R1.3 High level network protocol	Technology independent, high-level network protocol for allowing neutral access to domotic networks.
	R1.4 API	Public API to allow external services to easily interact with home devices.
R2 Automation	R2.1 Modeling	Abstract models to describe the house devices, their states and functionalities, to support effective user interaction and to provide the basis for home intelligence.
	R2.2 Complex scenarios	Ability to define and operate scenarios involving different networks / components.
R3 Intelligence	R3.1 Offline Intelligence	Ability to detect misconfigurations, structural problems, security issues, etc.
	R3.2 Online Intelligence	Ability to implement runtime policies such as energy saving or fire prevention.
	R3.3 Adaptation	Learning of frequent interaction patterns to ease users' everyday activities.
	R3.4 Context based Intelligence	Proactive behavior driven by the current house state and context aimed at reaching specific goals such as safety, energy saving, robustness to failures.

### 3.3 The DOG Domotic OSGi Gateway

The core component of an IDE is the software running on the additional embedded PC (either Residential or Home Gateway [23, 15]). In particular, in the COGAIN network, the DOG gateway has been studied for environmental control [4].

DOG (Domotic OSGi Gateway) is a domotic gateway designed to transform new or existing domotic installations into IDEs by fulfilling the requirements defined in Section 3.2. Design principles include versatility, addressed through the adoption of an OSGi based architecture, advanced intelligence support, tackled by formally modeling the home environment and by defining suitable reasoning mechanisms, and accessibility to external applications, through a well defined, standard API also available through an XML-RPC [26] interface. OSGi [22] is an Universal Middleware that provides a service-oriented, component-based environment for developers and offers standardized ways to manage the software life cycle. It provides a general-purpose, secure, and managed framework that supports the deployment of extensible service applications known as *bundles*.

DOG exploits OSGi as a coordination framework for supporting dynamic module activation, hot-plugging of new components and reaction to module failures. Such basic features are integrated with an ontology model of domotic systems and

sourrounding environments named DogOnt [3]. The combination of DOG and DogOnt supports the evolution of domotic systems into IDEs by providing means to integrate different domotic systems, to implement inter-network automation scenarios, to support logic-based intelligence and to access domotic systems through a neutral interface. Cost and flexibility concerns take a significant part in the platform design and we propose an open-source solution [5] capable of running on low cost hardware systems such as an ASUS eeePC 701.

3.3.1 DOG Architecture

DOG is organized in a layered architecture with 4 **rings**, each dealing with different tasks and goals, ranging from low-level interconnection issues to high-level modeling and interfacing (Figure 3). Each ring includes several OSGi bundles, corresponding to the functional modules of the platform.

**Ring 0** includes the DOG common library and the bundles necessary to control and manage interactions between the OSGi platform and the other DOG bundles. At this level, system events related to runtime configurations, errors or failures, are generated and forwarded to the entire DOG platform.

**Ring 1** encompasses the DOG bundles that provide an interface to the various domotic networks to which DOG can be connected. Each network technology is managed by a dedicated driver, similar to device drivers in operating systems, which abstracts network-specific protocols into a common, high-level representation that allows to uniformly drive different devices (thus satisfying requirement R1.1).

**Ring 2** provides the routing infrastructure for messages travelling across network drivers and directed to DOG bundles. Ring 2 also hosts the core intelligence of DOG, based on the DogOnt ontology, that is implemented in the House Model bundle (R1.2, R1.3, R2.1 and, partially, R2.2).

**Ring 3** hosts the DOG bundles offering access to external applications, either by means of an API bundle, for OSGi applications, or by an XML-RPC endpoint for applications based on other technologies (R1.4).

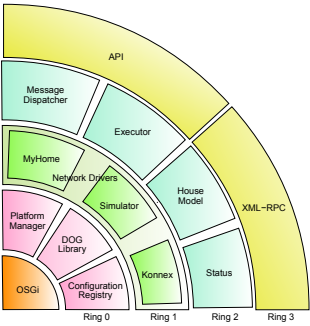


Fig. 3 DOG architecture.

3.3.2 Modeling in DOG

Modeling the house structure, its domotic components, their states and functionalities is achieved by defining a custom *ontology*, called **DogOnt** [3]. According to the classical Gruber’s definition [13] an ontology is an “explicit specification of a conceptualization,” which is, in turn, “the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold among them”. Today’s W3C Semantic Web standard suggests a specific formalism for encoding ontologies (OWL), in several variants that vary in expressive power [19].

DogOnt is a OWL meta-model for the domotics domain describing where a domotic device is located, the set of its capabilities, the technology-specific features needed to interface it, and the possible configurations it can assume. Additionally, it models how the home environment is composed and what kind of architectural elements and furniture are placed inside the home. It is organized along 5 main hierarchy trees (Figure 4), including: *Building Thing*, modeling available things (either controllable or not); *Building Environment*, modeling where things are located; *State*, modeling the stable configurations that controllable things can assume; *Functionality*, modeling what controllable things can do; and *Domotic Network Component*, modeling peculiar features of each domotic plant (or network). The *Build-*

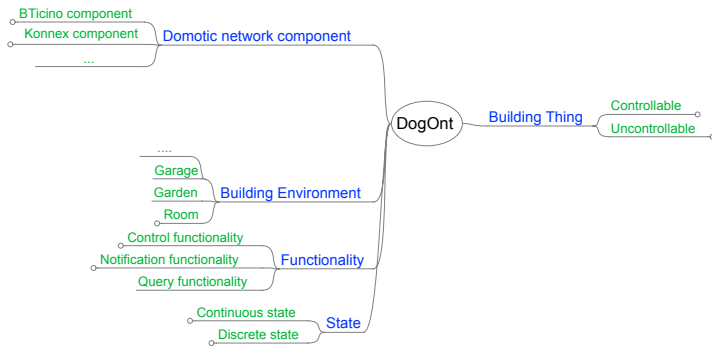


Fig. 4 An overview of the DogOnt ontology

*ingThing* tree subsumes the *Controllable* concept and its descendants, which are used to model devices belonging to domotic systems or that can be controlled by them.

Devices are described in terms of capabilities (*Functionality* concept) and possible configurations (*State* concept). Functionalities are mainly divided in *Continuous* and *Discrete*, the former describing capabilities that can be varied continuously and the latter referring to the ability to change device configurations in a discontinuous manner, e.g., to switch on a light. In addition they are also categorized depending on their goal, i.e. if they allow to control a device (*Control Functionality*), to query a device condition (*Query Functionality*) or to notify a condition change (*Notification Functionality*). Each functionality instance defines the set of associated commands

and, for continuous functionalities, the range of allowed values, thus enabling run-time validation of commands issued to devices. Devices also possess a state instance deriving from a *State* subclass, which describes the stable configurations that a device can assume. Each *State* class defines the set of allowed *state values*; states, like functionalities, are divided in *Continuous* and *Discrete*.

DOG uses the DogOnt ontology for implementing several functionalities encompassing **command validation** at run-time, using information encoded in functionalities, **stateful operation**, using the state instances associated to each device, **device abstraction** leveraging the hierarchy of classes in the controllable subtree. The last operation, in particular, allows to deal with unknown devices treating them as a more generic type, e.g., a dimmer lamp can be controlled as a simple on/off lamp. Ontology instances modeling controlled environments are created off-line by means of proper editing tools, some of which are currently being designed by the authors, and may leverage auto-discovery facilities provided by the domotic systems interfaced by DOG.

## 4 Eye-based Environmental Control

An intelligent domotic environment is of particular importance for an individual with any form of restricted mobility or other disability which can arise as a result of disease or aging. The facility to operate environmental systems and devices directly without the necessity of either having first to move physically to that device location and interact with it, or alternatively communicate to a carer their wish to have the device operated on their behalf, is important in achieving independent living.

In order for a disabled user to initiate a device operation there exists a wide range of user interface mechanisms (e.g., head movement switch, sip/puff switch) which can be tailored to the individual's needs and physical abilities. Usually the requirements of the individual are first assessed by a rehabilitation professional and an appropriate interface or range of interfaces for that person then determined and implemented. Suitable follow up of the individual over time then ascertains whether such interfaces are suitable. For some disabled users operating environmental systems by their eye movements is one such optional control mechanism and for others it is the only control mechanism available to them.

### 4.1 Overview of Eye Based Environmental Control

For some deteriorating health conditions the voluntary control of eye movements is the last controllable physical movement available. Such individuals can benefit from eye-operated assistive technology as part of a gaze-operated home automation system. A wide range of diseases and conditions give rise to an individual potentially having very restricted motor movement ability which limits them using other



interfaces and so eye based systems could be useful. The background to eye based control lies in the fact that humans exhibit a wide range of different types of eye movements which are usually not consciously controlled. However, saccadic eye movements can be voluntarily controlled and therefore can consciously and purposefully shift eye gaze, and therefore direct visual attention, to particular areas of the environment. These movements have long been studied to understand perceptual and cognitive aspects of numerous activities such as reading, examining pictorial displays or in vehicle driving behaviour.

In considering such saccadic movement behaviour, vision is construed as consisting of an alternating stream of very fast saccades (when little or no visual input takes place) interspersed with eye fixations of differing time lengths when eye gaze is almost stationary (although there are still micro-movements of the eye) at some spatial position. Usually the length of an eye fixation is indicative of the degree of visual processing taking place at that location coupled with determining where to fixate next in the environment. Because eye gaze can link directly to visual attention then it is feasible to use saccadic eye movements as an HCI control movement. However, eye gaze direction does not always relate to visual attention, which has been termed the Midas touch problem [17] which poses a particular difficulty when eye gaze is used to make a visual selection of a device control operation. Consequently eye based systems usually need some additional technique to indicate that an actual selection is required [16]. Virtually all systems that record saccadic eye movements are based on trying to determine the user's point of gaze as accurately as possible and then use this measured gaze location information to indicate a particular decision selection. More recently research has begun to investigate using the direction of the saccadic movements themselves and not just the gaze location [14].

Numerous commercial eye tracking systems exist which can be used easily by a disabled user. However, the cost of such systems remains comparatively high which limits their widespread adoption even though the technique has wide applicability. Research, particularly that supported through the COGAIN network, is gradually leading to a range of cost effective and affordable technological options (see [www.cogain.org](http://www.cogain.org)). Eye trackers for assistive technology usage typically fall into two camps: head mounted on the user, or remote and not attached to the user but positioned in front of them. Head mounted systems are small, lightweight, inconspicuous and allow the user to move their heads freely without any loss of eye data recording. Remote systems, on the other hand, are generally mounted in front of a user and record where the user is looking within a fixed range of measurements—generally circa 20-300 visual angle. These are ideally suited to a user interacting with a display positioned in front of them if the user is relatively immobile or else can be mounted on a user's wheelchair. With either type of system environmental devices can be represented on a monitor in front of the user as a symbol or in text and gazing at this can bring up a menu from which the user selects the particular action they require.

The advantages of such eye based approaches over other user control selection systems are that they can provide quick and effective direct control. Also they can be fairly intuitive in nature. In terms of limitations some systems can be difficult

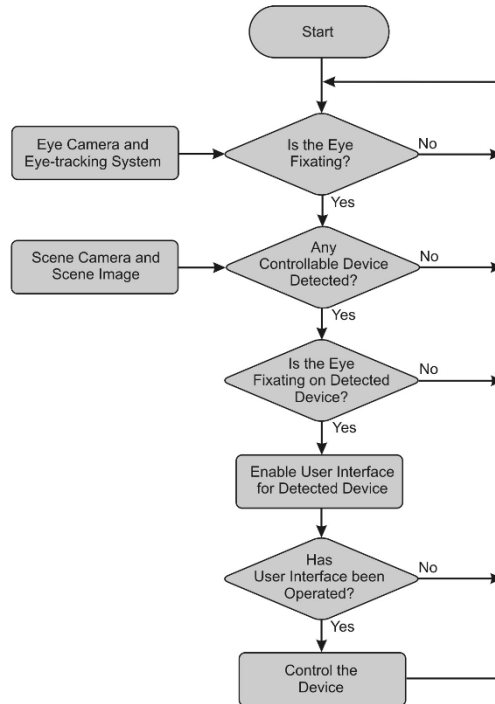
to set up for a particular disabled user but increasingly research developments are making them very user friendly (e.g. MyTobii).

## ***4.2 The ART system***

One innovative approach to eye based interaction control is illustrated through the ART (Attention Responsive Technology) system [12, 24]. In this, the overall aim is to allow the user to operate environmental systems simply by looking directly at a particular device which would then give rise to device operation. The concept is that this obviates the need for the user to first select which device they wish to operate from a displayed menu of potential devices. It was also a design requirement that the user would be mobile, for instance in an electric wheelchair, and so could approach different real world devices in their environment from various directions. To ensure the final system was fit for purpose input from potential end users was used at key developmental stages.

The system was developed using a head-mounted and a remote eye movement monitoring system separately to illustrate that either approach can be implemented and that both were feasible for a disabled user. Each system is coupled to a digital video camera which monitors the environment in front of the user (this is either a miniature camera mounted on a headband, if a head-mounted eye movement system is used, or else mounted directly on a wheelchair). Firstly, environmental devices which the user may wish to operate are imaged from various points of view by this camera and then a SIFT algorithm [18, 25] is run which results in suitably extracted and image processed SIFT features representative of the device which are then stored in a database. Additional or new devices can be added easily to the system by imaging these in a similar manner and their SIFT features automatically added to the database. Control operations for each device are also added so that when that device is recognised by the ART system then the appropriate control options can be presented to the user.

A user is first calibrated for the particular eye movement system in use and then the system works autonomously by constantly monitoring the user's eye movements and recognising whenever they are steadily gazing, using user defined fixation length criteria, at some point in their environment. The coordinates of where they are gazing are then calculated and related to the video camera image. A SIFT algorithm is then applied which analyses the camera image data around the point of gaze and determines whether a known device is actually being gazed at. If it is, then an interface is offered to the user specifically for that device alone which the user can decide whether to operate—this then overcomes any accidental device operation. Figure 5 shows a flow diagram of the system. The actual operation of the interface itself can again be by eye control. The approach has the advantage of not needing to present the user with an initial menu of all available devices to operate, instead offering a direct selection through eye gaze at the device itself.



**Fig. 5** Operational flow chart of the ART system.

In the developmental system several dedicated device interfaces have been constructed so that these can be presented to the user on a small monitor but these could equally well be presented either projected as an overlay onto, or beside, the actual device itself or in any other manner (e.g. audio) and format that the user could efficiently respond to. There is no real requirement for the user to be positioned in front of a monitor. Two system parameters are used to overcome possible false operation of a device simply because the user's gaze is recorded as falling upon it. Firstly, the user must gaze at a device for a pre-determined criterion time; this allows the software to identify the device in the scene camera image from the database of SIFT features as well as preventing the ART system trying to recognise other potential objects. Secondly, the user's eye gaze does not (of itself) initiate device operation but instead initiates the presentation of a dedicated interface just for that device. This permits a check on whether or not the user does in fact wish to operate the device. Figure 6 shows the developmental system in a laboratory with a user sitting in an electric wheelchair. The camera beside the user's head monitors the environment in front of them. In this set up the user's gaze is monitored by a Smarteye eye movement system ([www.smarteye.se](http://www.smarteye.se)) which uses three small cameras, coupled with

infrared light sources. For actual implementation, both the eye movement system and the environmental monitoring camera are mounted on the wheelchair itself.



**Fig. 6** Laboratory set up with user in a wheelchair

### ***4.3 Eye Based Environmental Control and Communication Systems***

The ART system is currently solely designed to facilitate direct operation of environmental devices. One next step is to incorporate other user requirements into its design. The approach represents just one way of utilising eye gaze as an assistive technology and deliberately set out largely to remove a user from being tied to their computer monitor. However, other approaches to using eye movements assistively, utilise this interactivity premise. In terms of recording eye gaze it is an easier proposition to have a user located directly in front of a monitor.

Employing this scenario different eye tracking techniques have been developed as have different software packages which facilitate communication, work, game playing and other leisure activities. Monitoring eye movements of a user whilst seated in a wheelchair can be used to provide real time precise steering and guidance [11] as well as for planning movement around an environment, before actually

moving, coupled with multisensory collision avoidance systems to prevent any accidents [21].

Whilst product development needs require that research targets specialised usages there is an overall need for eye based control to interface wholistically with the everyday living requirements of a disabled or elderly user, whether this be for environmental control or communication. Facilitating the user to move around their environment as well as control it is yet another exciting step.

## 5 Conclusions

This chapter provided an overview of the opportunities offered by domotic technologies and by eye tracking systems. Domotics is a new and promising way of implementing ambient intelligence and to enable ambient assistive living for elderly or disabled people. Eye tracking technology is essential for supporting mobility-impaired people and to enable them to use a computer system. The chapter presented the latest advances within the COGAIN project, where environmental control and eye tracking are coupled to provide a comprehensive system allowing user to remotely control all aspects of their home, by exploiting natural and intuitive gaze-based interactions.

**Acknowledgements** This work was supported by the European Commission Project 511598, Network of Excellence on Communication by Gaze Interaction (COGAIN). The ART project is supported by EPSRC grant EP/F008937/1. The authors also wish to acknowledge the significant contribution of Richard Bates, Emiliano Castellina, Dario Bonino to the results described in this chapter.

## References

- [1] Bates R, Spakov O (2006) D2.3 implementation of cogain gaze tracking standards. Deliverable, Communication by Gaze Interaction (COGAIN), IST-2003-511598, URL <http://www.cogain.org/results/reports/COGAIN-D2.3.pdf>
- [2] Bates R, Daunys G, Villanueva A, Castellina E, Hong G, Istance H, Gale A, Lauruska V, Spakov O, Majaranta P (2006) D2.4 a survey of existing ‘de-facto’ standards and systems of environmental control. Deliverable, Communication by Gaze Interaction (COGAIN), IST-2003-511598, URL <http://www.cogain.org/results/reports/COGAIN-D2.4.pdf>
- [3] Bonino D, Corno F (2008) Dogont - ontology modeling for intelligent domotic environments. In: 7th International Semantic Web Conference. October 26-30, 2008. Karlsruhe, Germany.

- [4] Bonino D, Castellina E, Corno F (2008) Dog: an ontology-powered osgi domotic gateway. In: 20th IEEE Int'l Conference on Tools with Artificial Intelligence, IEEE, USA
- [5] Bonino D, Castellina E, Corno F (2008) The DOG Domotic OSGi Gateway. <http://domoticdog.sourceforge.net>
- [6] Chung SL, Chen WY (2007) MyHome: A Residential Server for Smart Homes. Knowledge-Based Intelligent Information and Engineering Systems 4693/2007:664–670
- [7] Cook DJ, Youngblood M, Heierman EO III, Gopalratnam K, Rao S, Litvin A, Khawaja F (2003) Mavhome: An agent-based smart home. In: PERCOM '03: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, IEEE Computer Society, Washington, DC, USA, p 521
- [8] Davidoff S, Lee MK, Zimmerman J, Dey A (2006) Principle of Smart Home Control. In: Proceedings of the Conference on Ubiquitous Computing, Springer, pp 19–34.
- [9] DLNA (2007) DLNA overview and vision whitepaper 2007. Digital Living Network Alliance, URL [http://www.dlna.org/en/industry/pressroom/DLNA\\_white\\_paper.pdf](http://www.dlna.org/en/industry/pressroom/DLNA_white_paper.pdf)
- [10] Donegan M, Oosthuizen L, Bates R, Daunys G, Hansen J, Joos M, Majaranta P, Signorile I (2005) D3.1 user requirements report with observations of difficulties users are experiencing. Deliverable, Communication by Gaze Interaction (COGAIN), IST-2003-511598, URL <http://www.cogain.org/results/reports/COGAIN-D3.1.pdf>
- [11] Figueiredo L, Nunes T, Caetano F, Gomes A (2008) Magic environment. In: Instance H, Štěpánková O, Bates R (eds) Proceedings of COGAIN 2008
- [12] Gale A (2005) Attention responsive technology and ergonomics. In: Bust PD, McCabe PT (eds) Proceedings of the International Conference on Contemporary Ergonomics (CE2005), Taylor and Francis, 978-0-415-37448-4
- [13] Gruber T (1995) Toward principles for the design of ontologies used for knowledge sharing. International Journal Human-Computer Studies 43(5-6):907–928
- [14] Heikkilä H (2008) Gesturing with gaze. In: Instance H, Štěpánková O, Bates R (eds) Proceedings of COGAIN 2008
- [15] HGI (2008) Home gateway technical requirements: Residential profile. Tech. rep., Home Gateway Initiative, URL [www.homegatewayinitiative.org/publis/HGI\\_V1.01\\_Residential.pdf](http://www.homegatewayinitiative.org/publis/HGI_V1.01_Residential.pdf)
- [16] Instance H (2008) User performance of gaze based interaction with on-line virtual communities. In: Instance H, Štěpánková O, Bates R (eds) Proceedings of COGAIN 2008
- [17] Jacob R (1990) What you look at is what you get: eye movement-based interaction techniques. In: Carrasco J, Whiteside J (eds) Proceedings of the ACM CHI 90 Human Factors in Computing Systems Conference, Washington, USA, pp 11–18

- [18] Lowe D (2004) Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60, 2:91–110
- [19] McGuinness DL, van Harmelen F (2004) Owl web ontology language. W3C Recommendation, <http://www.w3.org/TR/owl-features/>
- [20] Nakashima H (2007) *Advances in Ambient Intelligence* Edited, Series: Frontiers in Artificial Intelligence and Applications, vol 164, chap Cyber Assist Project for Ambient Intelligence, pp 1–20
- [21] Novák P, Krajník T, Přeučil L, Fejtová M, Štěpánková O (2008) AI support for a gaze-controlled wheelchair. In: Istance H, Štěpánková O, Bates R (eds) *Proceedings of COGAIN 2008*
- [22] OSGI (2007) OSGi service platform release 4. Tech. rep., The OSGi alliance
- [23] Saito T, Tomoda I, Takabatake Y, Arni J, Teramoto K (2000) Home gateway architecture and its implementation. *IEEE Transactions on Consumer Electronics* 46(4):1161–1166
- [24] Shi F, Gale A, Purdy K (2006) Helping people with ict device control by eye gaze. In: Miesenberger K, Klaus J, Zagler W, Karshmer A (eds) *Computers Helping People with Special Needs*, Springer Verlag, Berlin, *Lecture Notes in Computer Science*, pp 480–487
- [25] Shi F, Gale A, Purdy K (2006) Sift approach matching constraints for real-time attention responsive system. In: Byun J, Pan Z (eds) *Proceedings of the 5th Asia Pacific International Symposium on Information Technology*, Hangzhou, China
- [26] Winer D (2003) XML-RPC specification. Tech. rep., UserLand Software