# E-servant: an intelligent, programmable system to support and integrate assisted living technologies

*Richard Picking[1], Vic Grout[1], Roberto Casas[2], Ruben Blasco[2]*

*[1]Department of Computing, Glyndwr University, Wrexham, LL11 2AW, Wales, UK*
*[2]HOWLab, Universidad de Zaragoza, Zaragoza, Spain*
*E-mail: r.picking@glyndwr.ac.uk*

Published in Healthcare Technology Letters; Received on 18th February 2014; Revised on 3rd April 2014; Accepted on 3rd April 2014

The ‘E-servant’, a programmable system to control and manage assistive technologies, telehealth and telecare devices in a home environment is presented. The E-servant is programmed using a simple graphical interface that allows the user to build a dialogue in the form of a production rule system, which is triggered by a patient- or technology-initiated event. The patient interacts with the system through a personalised user interface to reach their goal of completing a task. These tasks, which the authors call ‘scenarios’, can be designed for users of different abilities (cognitive and/or physical). They can also be given priority levels, for example if a potential emergency situation arises in the patient’s home, a scenario associated with the sensing of this event takes highest priority. The research presented in this Letter outlines the E-servant, its programming tool and reports its evaluation in living laboratory settings. The results suggest that it can be used as a central management system for supporting an integrated support environment for facilitating healthcare and activities of daily living, especially for older patients.

**1. Introduction:** The development of the E-servant was initially motivated by the now well-documented problem of ageing populations, and the recognition that people will need support to remain independent for longer into their old age. To put this into perspective, European Union population projections suggest that the ratio of people aged 65 years or over will increase from 17.1% to 30.0% in 2060 (from 84.6 million in 2008 to 151.5 million people in 2060) [1]. Similar figures can be found in the USA, where older people will represent 20.2% of the population in 2050, or in Japan, with 39.6% [2, 3].

Older people may suffer several physical and/or cognitive impairments which increase over time. Old age affects sensing, information processing capability, reduces speed and increases timing of precise movements. All these issues increase difficulties of comprehension of complex scenarios which may require varying degrees of multi-tasking. As a consequence, older people are most vulnerable to accidents, particularly in their homes [4].

To alleviate such problems, assistive technologies are becoming more and more widely used, and they are also becoming more sophisticated (for example telehealth equipment). However, every individual is different and may have differing needs, hence the requirement for these technologies to be adaptable and personalised where possible. This requirement comes with its own difficulties, as configuring and integrating potentially complex devices into a person’s daily routine may be time-consuming, expensive and error-prone.

The E-servant attempts to support healthcare professionals in such situations, enabling them to program devices in an intuitive, visual manner, with no need for computer or device programming skills. The strategy of using visual programming for this task has been successful on other similar projects described in [5], although none of them support the adaptability offered by the E-servant.

**2. Methodology:** The design of the E-servant conforms to a cognitive architecture paradigm, specifically following a problem space model (PSM) approach, first proposed by Newell and Simon [6]. This methodology has been used in a number of research projects involving the design of artificial intelligence agents (AIAs), and has been successful in predictive analysis, where user interaction is simulated by the AIA, most famously in aircraft pilot simulation [7]. We have adapted this methodology to design an intermediary agent called the E-servant, which behaves as closely as possible to a human intermediary, in terms of its interaction with the user.

The problem space is defined in detail in the form of a production rule system, initiating as an event which triggers a scenario, where the user solves a problem (interacting with the E-servant, the sensors/actuators, and the appliances). An example of this follows:

1. Initial event: the E-servant identifies that an item of food in the refrigerator has passed its use by date. It notifies the user.
2. The user can acknowledge the notification via the user interface, or if required, can seek further help.
3. If the user seeks help, another help scenario is initiated – this demonstrates the PSM approach of enabling scenarios to stack, so that they can be used in an intelligent way for individual user needs. Once the help scenario is completed, the original problem scenario is continued.
4. When the item is removed, or the user indicates satisfaction with the issue, the scenario closes, and the E-servant awaits another scenario triggering event.

The adaptivity of the E-servant is based on a formalised set of user models, which act as defaults for the E-servants’s knowledge of differing human characteristics and abilities. These defaults are modified according to users’ actual interaction with the E-servant and their behaviour in the general environment of the home. A personas methodology [8] was used for this, where typical categories of user were defined, based initially on research derived from a survey of elderly independently living people. Deviation from the personas is informed by an automated report [which we call quality of life evaluation (QoLE)] generated by the intelligent real-time monitoring of interaction and user performance.

**3. E-servant architecture and functionality:** The E-servant is an embedded computer that centralises the communication with devices, sensors and interfaces, as well as behaving as a gateway to the outside world. The E-servant is essentially the central hub of the whole system that, being aware of the context and user, enhances its intelligence. It is also a learning system, which detects and compensates the behaviour, habit changes and loss of abilities of the user. Sensors using radio-frequency identification (RFID), ZigBee, powerline communication (PLC), Ethernet, Wi-fi, Bluetooth and infrared technologies all enable the system to interact with the home environment.

For each user interaction, the E-servant assists the user taking into consideration the user capabilities and environmental context. Also, the E-servant continuously checks the status of the devices (for example, kitchen appliances), providing warnings through its user interfaces (for example, smart television, tablet PC) if there is any problem or event to be notified (for example, the fridge door is open or the cooking has finished). The E-servant is also able to detect emergency situations in the home, automatically taking corrective actions if the user does not respond. Finally, the E-servant records relevant events that have occurred in the home. These data are processed and analysed using artificial intelligence methods to extract findings about the cognitive level of the person. This could be useful to carers and/or relatives. This information is used to create an expert service which we call QoLE. The QoLE generates a detailed activity report which is forwarded to the user's carers or relatives. The QoLE acts as a tool to determine whether the patient's user profile (and hence user experience) needs to be modified.

To program the E-servant for individual patients, the user (for example, healthcare professional) will interact with a simple graphical user interface, which supports a drag and drop approach to the building of scenarios. An example screen is shown in Fig. 1.

The E-servant's production rule protocol follows an event, rule action cycle, which is read from an XML representation of each scenario. In all, there are six constructs that can be combined together in a hierarchy to represent an executable scenario: start event, rule, action, timer, response and termination event (Fig. 2).

The visual programming of a scenario begins with the user creating a start event (left-hand side of screen in Fig. 1). Its attributes can then be populated by typing their values on the computer keyboard (for example the scenario's name, or its level of priority). The scenario is then 'built' by further dragging and dropping of the constructs onto the scenario timeline. Each construct has programmable attributes that determine its behaviour. Where there is more than one consequence of an action, the scenario's tree branches to enable it to follow a different path. Scenarios can terminate at the end of any branch (as depicted by the termination event construct in Fig. 1 and 2. Throughout the scenario creation process, there is no requirement for the user to enter any 'program code'.

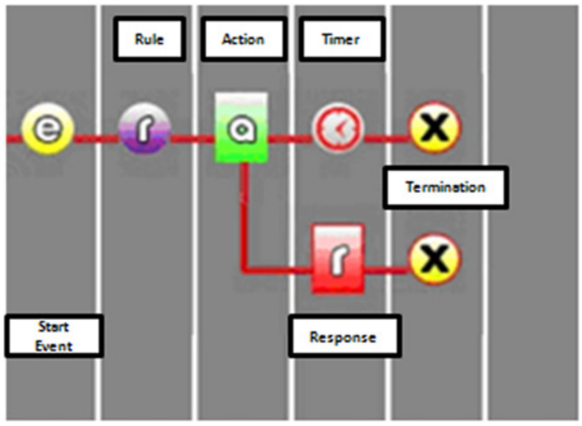
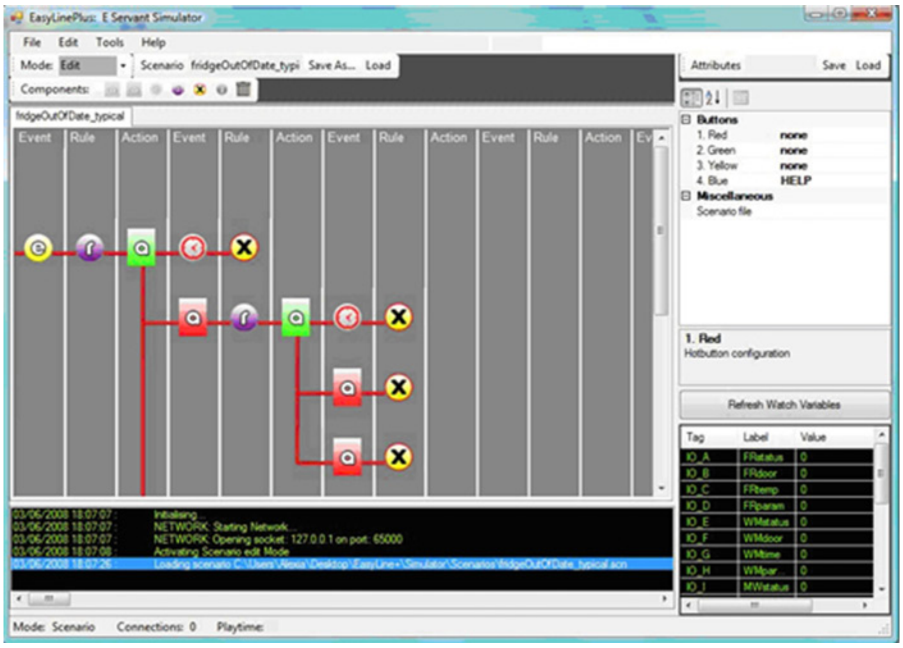


**Figure 2** *Overview of the six E-servant constructs. Constructs (the icon-like objects) are dragged onto the timeline and connected together to form a tree structure that becomes a full scenario*

The E-servant communicates with external devices using an XML message protocol. The first message in a scenario is usually sent by the E-servant to an external device. Subsequently, the E-servant expects one of several responses back from the external device, such as an acknowledgement from a user interface, or a change in status of an appliance. A timeout mechanism is provided to deal with the case where no response is obtained. There are two key message types used: action messages and response messages. Both these messages inherit the same common attributes.

**4. Example scenario:** An example E-servant dialogue is now explained. It is translated into an English language representation here for clarity, rather than the XML used by the system. A typical screen display of an event in the scenario (level 2) is shown in Fig. 3. In this case a smart television interface was used, but the technology supports a wide range of mobile and static devices.



```
03/06/2008 18:07:07 :      Initialising...
03/06/2008 18:07:07 :      NETWORK: Starting Network...
03/06/2008 18:07:07 :      NETWORK: Opening socket: 127.0.0.1 on port: 65000
03/06/2008 18:07:08 :      Activating Scenario edit Mode
03/06/2008 18:07:26 :      Loading scenario C:\Users\Alexia\Desktop\EasyLine+\Simulator\Scenarios\fridgeOutOfDate_typical.scn
```

Mode: Scenario   Connections: 0   Playtime:

Buttons
| | |
|---|---|
| 1. Red | none |
| 2. Green | none |
| 3. Yellow | none |
| 4. Blue | HELP |

Miscellaneous
Scenario file

1. Red
Hotbutton configuration

| Tag | Label | Value |
|---|---|---|
| IO_A | FRstatus | 0 |
| IO_B | FRdoor | 0 |
| IO_C | FRtemp | 0 |
| IO_D | FRparam | 0 |
| IO_E | WMstatus | 0 |
| IO_F | WMdoor | 0 |
| IO_G | WMtime | 0 |
| IO_H | WMpar... | 0 |
| IO_I | MWstatus | 0 |

**Figure 1** *E-servant programmer interface. Graphical canvas shows a scenario that is currently running. This interface can be used to monitor patient–system interaction, and it is also used as the visual programming tool*

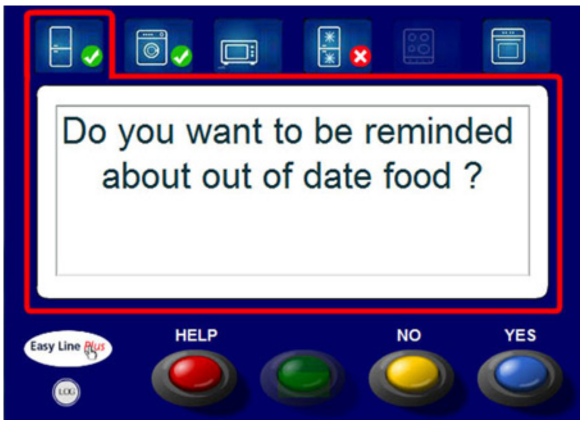


**Figure 3** *Example of patient–smart fridge dialogue managed by the E-servant. Appliances that can be controlled/monitored are shown at the top of the screen. Messages (which are also articulated audibly for certain user profiles) are shown in the middle of the screen, and user interaction is performed by pressing one of the four coloured 'teletext' buttons on the TV remote control unit*

FRIDGE Scenario (ID = 4): 'Food about to expire in the fridge': *Situation:* A daily check is performed to detect food contained in the fridge that is about to expire. A notification message (event) is sent to the user.

User profile: Expert (LEVEL3)

*Level 1*
E-servant: 'Some food is nearly out of date. See LIST?'

User response 1: 'HELP' (Goes to help scenario)
User response 2: 'NO' (Goes to level 2)
User response 3: 'YES' (Terminates scenario and goes to LIST mode)
User response 4: Nothing (if the button is not pressed, the timeout will terminate the scenario anyway)

*Level 2*
E-servant: 'Do you want to be reminded?'

User response 1: 'HELP' (Goes to help scenario)
User response 2: 'NO' (Goes to level 3a)
User response 3: 'YES' (Goes to level 3b)
User response 4: Nothing (if the button is not pressed, the timeout will terminate the scenario anyway)

*Level 3a*
E-servant: 'OK. I'll leave it to you to do it.'

User response 1: 'HELP' (Goes to help scenario)
User response 2: 'OK' (Goes to main menu)
User response 3: Nothing (if the button is not pressed, the timeout will terminate the scenario anyway)

*Level 3b*
E-servant: 'OK. I'll remind you later if you forget.'

User response 1: 'HELP' (Goes to help scenario)
User response 2: 'OK' (Goes to main menu)
User response 3: Nothing (if the button is not pressed, the timeout will terminate the scenario anyway)

User profile: Standard (LEVEL2)

*Level 1*
E-servant: 'Some food is nearly out of date. See LIST?'

User response 1: 'HELP' (Goes to help scenario)
User response 2: 'NO' (Goes to level 2)
User response 3: 'YES' (Terminates the scenario and goes to LIST mode)
User response 4: Nothing (if the button is not pressed, the timeout will terminate the scenario anyway)

*Level 2*
E-servant: 'OK, I'll remind you about food about to expire in the fridge.'

User response 1: 'HELP' (Goes to help scenario)
User response 2: 'OK' (Goes to main menu)
User response 3: Nothing (if the button is not pressed, the timeout will terminate the scenario anyway)

**5. Evaluation:** The E-servant has been evaluated by 63 end users and 31 formal and informal careers in two living labs situated in Spain (University of Zaragoza, Zaragoza) and UK (Glyndŵr University, Wrexham). The Zaragoza lab is a residential apartment that has been configured with the smart home technologies controlled by the E-servant. In Wrexham, a usability lab was converted to a living space with the exact same configuration, although in both cases, they were not occupied for more than one day by the evaluators. Of the end-user evaluators, 11 (8 female) were younger than 60 years old, 45 were 60-79 (28), and 7 were 80 or older (4). Participant disabilities included visual impairment (13), hearing (13), cognitive (12) and motor (23). During the evaluations, a number of scenarios were 'played out' over several hours. These were as follows:

*Scenario 1:* 'Coming home from shopping'. The participant comes home from shopping and he/she is asked to store all the

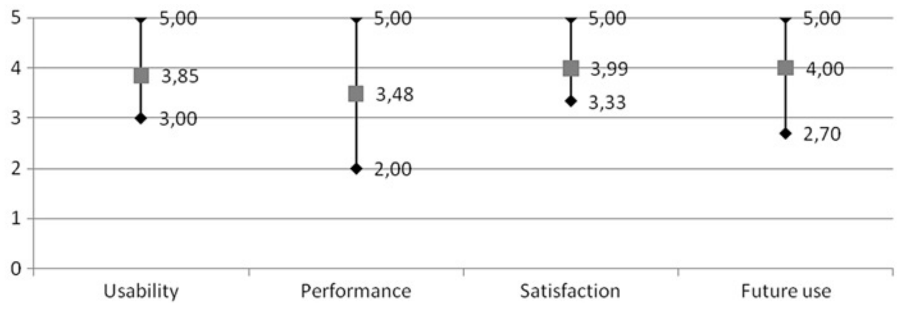


**Figure 4** *Summary of user assessment of E-servant operation, displaying means and variance. A total of 32 questions comprised the assessments, as well as qualitative data in the form of comments and opinions*

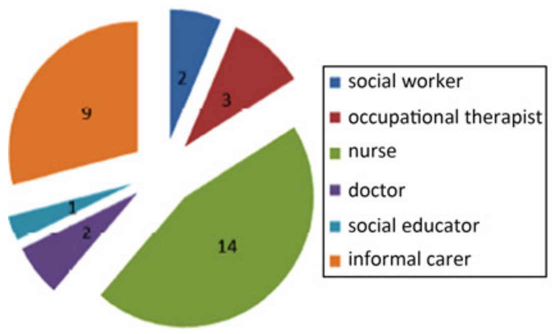


**Figure 5** *Carer group background*

items from the shopping bag into the fridge, freezer or cupboards. The E-servant advises that one of the items is out of date.

*Scenario 2:* 'Making dinner'. The participant is asked to show a frozen pizza to an RFID reader. The E-servant identifies the item and asks the user if he/she wants to cook it. If yes, it goes to the 'set_oven_configuration' with the relevant parameters. Then the participant goes to the living room (maybe to watch TV) and is informed when the food in the oven is ready.

*Scenario 3:* 'Doing the laundry'. The participant is asked to do a laundry task (simulated through the spin program to avoid long waiting times) using different clothing items.

*Scenario 4:* 'My house is on fire'. The participant is watching TV while the hob is on. The smoke detector is triggered simulating an emergency.

During and following the tests, assessments of usability, performance, satisfaction and likelihood of future adoption were made. The end user assessments were positive with an average score of 3.85 on a 1–5 semantic rating scale (Fig. 4).

The carer evaluation group comprised a mix of professions, as depicted in Fig. 5.

This group experienced the E-servant from both the perspective of the end-users, and from its configuration and management, including the programming of individual scenarios. Overall, they evaluated the system with a mean of 3.35 on a 1–5 semantic rating scale. As a tool to detect 'routine changes' in end-user behaviour, it was scored highly at 4.33, and the adaptability of the E-servant (including the carers' ability to change the scenarios programmatically) scored 3.35.

Further informal demonstrations to healthcare professionals including occupational therapists, palliative care specialists, community nurses, general practitioners and hospital consultants were also successful.

**6. Conclusion:** The E-servant has demonstrated much potential in facilitating the simple creation of interactive dialogues that can be personalised for individual users. The resulting user experience it supports has been evaluated to be broadly positive by both patients and carers. We propose that it can be adapted for a wide range of telecare and telehealth technologies in addition to those that support activities of daily living, although we have yet to perform a comprehensive analysis of potential correlations between user profiles and evaluation data. These anonymised data for each participant in our studies are available online [9].

The visual programming of the E-servant, although straightforward, does need careful consideration so as not to potentially confuse the patient as an end user. Consequently, we recommend that trained carers should be responsible for performing that task. Our research also identified that automatic adaptation of the patient interface is unwise, as predictability and consistency are critical factors. Instead, the E-servant's QoLE provides advice to professionals and informal carers as to whether the current user profile is appropriate.

**7. Acknowledgments:** This research has been supported through the European Union Framework Six (FP6) Information Society Technologies (IST) programme (No. 045515) EASYLINE+, 'Low Cost Advanced White Goods for a Longer Independent Life of Elderly People'.

## 8 References

[1] Giannakouris K.: 'Ageing characterises the demographic perspectives of the European societies', *Eurostat, Statis. Focus*, 2008, **72**, pp. 1–11
[2] Resident Population Projections by Sex and Age: 2010 to 2050. Available: http://www.census.gov/, accessed February 2014
[3] NIPSSR: 'Population Projections for Japan: 2001–2050. With Long-Range Population Projections', 2002, pp. 2051–2100
[4] Angermann A., Bauer R., Nossek G., Zimmermann N.: 'Injuries in the European Union. Statistics Summary 2003–2005' (Austrian Road Safety Board, Vienna, Austria, 2007)
[5] Maternaghan C., Turner K.J.: 'Programming home care', in Wolters M.K., Turner K.J., Lakany H. (Eds.): 'Proc. advances in techniques and technologies assisting care at home' (IEEE Computer Society, Los Alamitos, California, USA, 2011), pp. 5.1–5.7
[6] Newell A., Simon H.: 'Human problem solving' (Prentice-Hall, 1972)
[7] Jones R.M., Laird J.E., Nielsen E., Coulter K., Kenny P., Koss F.: 'Automated intelligent pilots for combat flight simulation', *AI Mag.*, 1999, **20**, (1), pp. 27–42
[8] Cooper A.: 'The inmates are running the asylum' (Sams Publishing, 1999, 1st edn.)
[9] Easyline+: 'D7.2 Report with the end user tests+D7.2 ANNEX', Available: http://www.easylineplus.com/ (accessed April 2014)