Virtual Butlers! A technology so close and so far!
Insights from the RASCALLI project

Brigitte Krenn, Jörg Irran, Gregor Sieber, Marcin Skowron
Austrian Research Institute for Artificial Intelligence
Freyung 6, 1010 Vienna, Austria

1. Introduction

Today humans benefit from extending their knowledge making use of the Internet, various databases and other sources of digital information. They gain access from different locations using personal computers, laptops, mobile phones, PDAs etc. Various ways to communicate and to exchange information with others are supported by community platforms. The tools/applications we currently employ, however, are all fragmented. The laptop does not know anything about the desktop computer and vice versa. The same holds for the mobile. In the best case, the desktop computer holds a backup of the important data of the laptop of its user, and the other way round, and the address database of one's mobile and one's personal computer are synchronized. This requires the user to actively take care of it. More and more people become members of internet communities and platforms such as Xing, Facebook, Delicious, etc. Each one requires its own login and personalization. Thus a single individual has many web appearances which do not know from each other and need to be maintained separately. Only recently, programs have become available that can be customized by the user to log onto the user's various sites with a single password. See OpenID or kwalletmanager (http://openid.net/, http://en.wikipedia.org/wiki/OpenID, http://utils.kde.org/projects/kwalletmanager/) for browser-based online or offline password management systems.

What is missing from a conceptual point of view is a kind of control unit, a component, let's call it Butler, that

(i) knows of the different devices and virtual appearances of a user,

(ii) is capable of accessing data from the various devices and sites,

(iii) is able to transform the data into knowledge, in order to proactively support the user in fulfilling her tasks and pursuing/reaching her goals.

Technically, it is important that new devices can be integrated with the control component, let's call it "Butler". The user not only wants to bundle her data distributed over PC, laptop, mobile phone and various web platforms, but also desires to connect the remote controls of various household appliances and health care components to the system, and the user wants to do it anytime anywhere. Ideally, a Virtual Butler would not only take care of one's virtual presences and the devices at home, but also negotiate with outside infrastructures such as lighting and air conditioning of hotel rooms etc.

Obviously the components connected to a Butler will be upgraded and exchanged over time. New functionality will be added. In order to be suitable for the masses, such a Butler system must be easy to maintain. Connecting a new tool or updating/upgrading an already existing one ideally should not require any re-programming that goes beyond formal interface definitions. More importantly, an interface is called for that allows the Butler to "grasp" the functionality of new devices/tools, compare it to that of previously known ones, and integrate to new ones into the Butler's current knowledge structure. By receiving the interface instructions in an "understandable" way, the Butler will be able to use devices/programs to the benefit of the user without requiring explicit intervention from the user. One possibility for tools to make known their functionality to the Butler is by including an ontology that functions as an instruction leaflet that describes what the individual tool can be used for, i.e. the concepts and relations it "understands" for an input and the output it provides, as well as how input and output are conceptually related. Apart from the problems of ontology merging (that might be overcome by developing standards for the basic knowledge of Butlers and Butler compatible tools), an important issue here is that purely symbolic knowledge will not be enough for the Butler to function properly. It will be important to associate symbols from the ontology with experiences or episodes.

One possibility of how Butlers can ground their knowledge is learning via self-experience, in particular, research into affordance-based learning in robotics. See for instance [Irran et al, 2006]. Following this approach, the psychological concept of affordances is utilized and transformed to an experience-based model of self-learning about the environment and interaction capabilities the individual butler is designed for [Irran et al, 2007]. The crucial issue here is to reduce data complexity. Thus engineering of the input data and perception channels is important as this directly influences the training of the classifiers.

Another approach is to utilize insights from research into human memory structuring for modelling the long-term and working memory for the Butler. The crucial issue here is how findings of cognitive theories can be transformed into computer programs that are able to handle aspects of the Butler's mind in a way that is flexible enough to deal with computer generated knowledge with its poor semantics and to scale to different domains and data volumes. Up to date implementations of cognitive models are still on an exemplary level where much fine-tuning is required for rather narrow tasks, and scalability is still an open issue.

In this paper, we present developments towards a Virtual Butler technology we have stared to model and implement within the RASCALLI project, illustrating how close and how far we are today from a Butler technology as sketched in "Tina and her butler". RASCALLI (Responsive Artificial Situated Cognitive Agents that Live and Learn on the Internet http://www.ofai.at/rascalli) is a European Cognitive Systems project which aims at developing virtual companions (Rascalli) to support the human user in retrieving information from the Internet and other digital information sources in a personalized way. Through interaction between user and system, the Rascalli learn about the interests and preferences of their users and apply this knowledge to find information from the Internet and other digital knowledge sources that fit the users' needs.

2. Rascalli action-perception architecture

To realize an interaction based knowledge acquisition architecture for virtual environments, a virtual embodiment for the Rascalli was created. The agents are equipped with a collection of sensor channels geared towards the particular environment, and a set of specialized software tools (actions) through which they interact with the environment. In the scope of the RASCALLI project, various sets of action-perception tools as well as knowledge acquisition, reasoning and decision making mechanisms are designed, implemented and tested. See Figure 1 for the organization of the architecture of individual Rascalli. Due to its modularity and flexibility, the RASCALLI development platform eases the creation of individual virtual agents via assembly from a set of existing software components [Krenn, Schollum, 2008]. Different realizations of the system (including various implementations of a cognitive component) can be assembled and evaluated to find an optimal set of system components for a given objective.

In the following, we briefly introduce the layers of the Rascalli architecture their relevance for a Butler system.

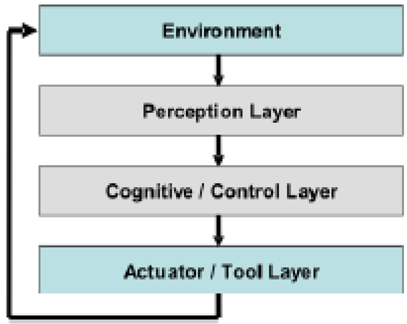

Figure 1: Rascalli agent architecture

Rascalli perception

The aim of the perception layer is to provide the individual Rascallo with the capabilities to perceive an input situation and to distinguish a set of features necessary for selecting an appropriate action (tool application), based on the similarities between input situations and the episodes, which the agent encountered before. The current environment includes the Internet, databases, knowledge resources, the user and other Rascalli. The Rascallo perceives its environment via a set of perception sensors, implemented as software tools as a set of unique entities with their own characteristics and properties. These include strings of written language originating from the user or extracted from HTML documents, markup tags, meta-data information about binary files, information about the accessibility of various Internet and local tools and resources, user feedback, etc. In their current realization the Rascalli are equipped with an input processing component that incorporates methods and tools from natural language processing, information retrieval and extraction, knowledge representation and modeling. They represent the Rascallo's sensor channels and thus categorize and restrict/filter the agent's capabilities to perceive its environment.

Rascalli actions

The Rascalli agents' current set of actuators are geared towards performing actions necessary to assist the user in accessing information from the Internet and domain-specific knowledge bases, and actions necessary to communicate the findings to the user. The actions are realized as application of software tools the agent is equipped with to data available in the environment such as input from the user, web pages and knowledge bases accessible to the individual Rascallo. Accordingly, the agent's perception is also realized as a collection of tools actively operating on the Rascalli environment. This approach relates to active perception as known from robotics, i.e. that perception is not passive, but actively influenced by action [Pfeifer, Scheier, 1999, pp. 431f.]. For a more detailed description of the Rascalli action and perception tools see [Skowron et al., 2008].

The cognition layer

To achieve intelligent assistance capabilities, a Rascallo must have abilities to select appropriate actions in a given situation. It has to be able to find cues in an input situation that can be related to one or more possible action applications. In the current system, the cognitive layer supports the following approaches for coupling perception and action, and action selection: a rudimentary rule based action selection mechanism, a cognitive model-driven memory organization and action selection, a data-driven and classification-based action selection.

Depending on the developmental stage of an individual Rascallo (i.e. the realization of its cognitive layer), tools are selected and executed arbitrarily, motivated by drives, or deliberately chosen based on the given input (perception layer) and previously made experience. The approaches we experiment with in realizing the cognitive layer are discussed in more detail in the following section.

3. Three realizations of the Rascalli cognitive layer

The RASCALLI platform allows the usage of and the research into various, cognitively motivated approaches to action selection ranging from a low level data-driven (bottom-up) approach where an agent acquires knowledge based on logs from its interaction with the user and the other entities from its environment, to theory driven (top-down) approach based on the DUAL/AMBR cognitive model [Kokinov, 1994]. The different approaches have specific characteristics and predispositions in the scope of their applications. While the bottom-up approach is more flexible and robust, the theory-driven approach provides more potential for fine-tuning and modeling accuracy for selected examples from restricted domains.

In the following, we present the approaches realized so far in the RASCALLI system.

Rule-based action selection

The rule based action selection component is a trivial implementation of a control component, based on hard-coded action selection rules primarily used for sandbox-like testing. The rules match to specific cues in the input data arriving from the sensor channels. Relevant information is extracted and passed on to the appropriate actuator tool. Even though seemingly non-trivial behavior can be accomplished through a series of interactions of the Simple Mind and the available tools, the Simple Mind does not contain any cognitive aspects such as memory or learning.

Memory Structuring using the DUAL/AMBR cognitive model

The top-down action selection mechanism currently realized in the RASCALLI platform implements the DUAL/AMBR cognitive architecture [Kostadinov, Grinberg, 2008]. It includes a long term memory (LTM) where general and episodic knowledge is stored, and a working memory (WM) constituted by the active part of LTM, perceptual input and goals. The DUAL mind operates only on represented knowledge and has only a mediated connection to the body and the environment. Thus it contains a partial, selected representation of the environment at an abstract conceptual level and experiential memories related to specific episodes of interaction of a Rascalli agent with its environment.

Data-driven knowledge acquisition and action selection

With the bottom-up knowledge acquisition, we aim at a robust mechanism which autonomously endows the Rascalli with the knowledge necessary to perform its tasks. Since the agent deals with a dynamic environment, it is insufficient to provide ex-ante (e.g. through human expert knowledge) a full spectrum of knowledge required by the agent to perform its wide range of tasks and to adapt to the ever-occurring changes. Therefore the Rascalli are also equipped with a learning mechanism that is sensitive to the active perception of the environment, including the user activities within the system. The selection of a particular action is based on the similarities with other actions the agent had successfully performed in the past, i.e. received positive feedback from the user. The action selection classifiers are implemented as Maximum Entropy models [Jaynes, 1957]. The data used for training the classifiers represents an input situation in terms of entities perceived by the agent and an action associated with it.

4. Rascalli to user communication

A selection of user interfaces has been implemented for the Rascalli to communicate with and present information to the user. The Multi-Modal Generation Component provides a middleware functionality between the output of the cognitive layer or tool output and the user interfaces. The generation component implements a template-based approach by encoding vocabulary, phrases, gestures etc. combined with the output of the Rascalli tools and context data. For the technical realization, we use the Velocity template generation engine (http://velocity.apache.org/) as it allows to design and refine templates separately from the application code. An interface manager keeps track of what interface the user is connected to. The multi-modal generation component uses different methods to present data, depending on the capabilities of the interface device.

In the following, introduce the currently implemented Rascalli interfaces.

Embodied Conversational Character (ECA)

With the ECA interface (Figure 2) we have provided a personified interface where the Rascalli system presents itself in a human-like conversational manner, however restricted. While the user input is constrained to free text input into a window and pressing a praise and a scolding button (thumbs up/thumbs down) to reduce input processing errors, the system presents its output to the user via multi-modal, verbal and nonverbal behavior. Whether an ECA interface to the Rascalli is beneficial for the user still needs to be explored in user tests. Results from early user tests with the Rascalli ECA interface have shown that users like communicating with the ECA. Whether this initial positive attitude will hold over a longer period of Rascalli usage and with the availability of more standard web interfaces to the Rascalli system still needs to be explored.

Web Interface

The web interface provides a direct interface to some of the data available in the Rascalli system, and allows direct tracking of user behaviour, avoiding language analysis issues which could lead to mistakes in learning of user preferences or task solving strategies. Examples from the Rascalli Web interface are given in Figure 3, depicting pages for user registration and Rascallo/a creation, user login, user-driven specification of input to the Rascallo, history of Rascallo-user communication.

Jabber/XMPP chat client interface

Finally, with the Jabber interface (www.jabber.org), we provide a generic text communication interface to the Rascalli (Figure 4). They can be part of a user's contact list just like any human chat partner. Since there are many clients available for the Jabber system (including ones for mobile phones), it provides a ubiquitous and un-inhibitive connection to the agent. This interface makes it easy to implement Rascalli presences on various mobile devices. Moreover, it is also possible to implement a communication protocol on top of the Jabber connection to pass on sensory data from whatever sensors are available in the mobile device, and thus can help the agent assess and interpret the current situation the device is used in.

Figure 2: The Rascalli ECA interface; the user may chose a male (Rascallo) and a female (Rascalla) character

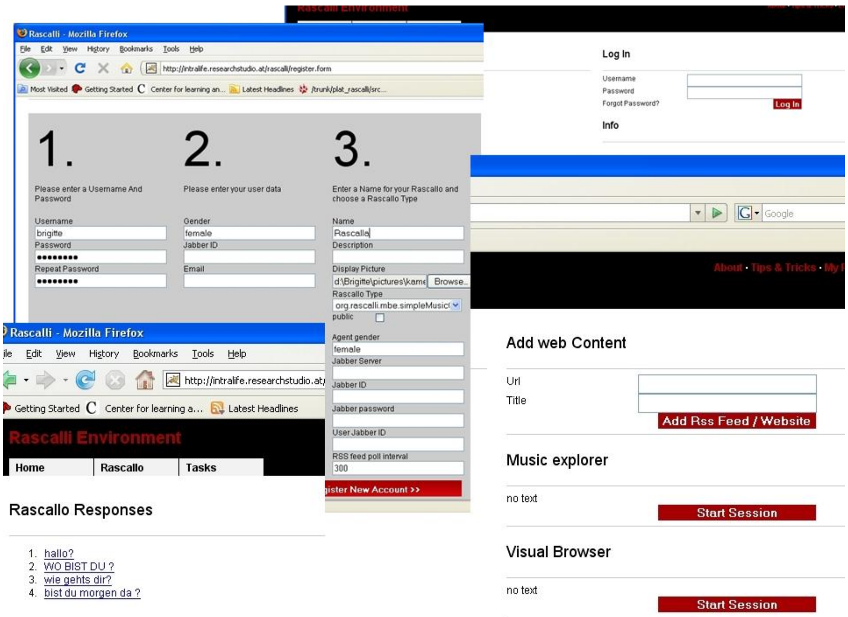

Figure 3: The Rascalli Web interface: examples for user registration and Rascallo/a creation, user login, user-driven specification of input to the Rascallo, history of Rascallo-user communication

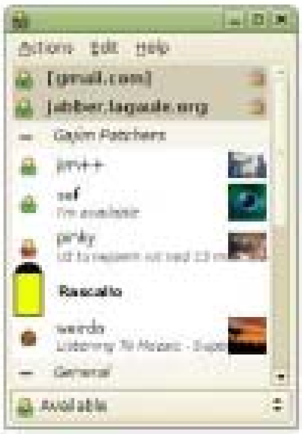

Figure 4: The Jabber client interface

5. Conclusion

Summing up, we have presented three contributions from the RASCALLI project, which we view as starting points towards the realization of Virtual Butlers. These are multiple user interfaces, action selection, and tool integration.

Multiple User Interfaces

Within the RASCALLI project, we have gained experience in implementing agents that live on a local platform, but are able to switch between interfaces as required by the situation and the preferences of the user. In the RASCALLI system, this is accomplished by using an abstraction layer, the multi-modal generation component, to generate system output behaviour that is adapted to the device used. I.e., the Virtual Butler itself resides on a local platform (at home or at a server farm) and uses several interfaces to communicate with the user, registering and remembering where the user is and what interface she is currently using. This is where we are now.

Talking about the future, the Virtual Butler moves (as in the concept of a mobile agent), with all its data, into the different devices used in the system. This could be an advantage if there is e.g. no internet connection to the original server available, but the VB can still be used to take control of local devices (e.g. making coffee according to the user's preferences on a coffee machine connected to a local wireless network) or assist in personal information management (keeping a schedule, reminding the user of meetings, assist in making calls, etc).

Action selection

To achieve intelligent assistance capabilities, a Butler must have abilities to select appropriate actions in a given situation. It has to be able to find cues in an input situation that can be related to one or more possible action applications. The examples of the action control components implemented for the Rascalli include a top-down driven cognitive architecture and a bottom-up machine-learning-based knowledge acquisition and classification-driven action selection. The different approaches have specific characteristics and predispositions in the scope of their applications.

For the Virtual Butler scenario, the ability, we have provided with the RASCALLI system, to integrate and evaluate the bottom-up approach with the theory-driven approach in a single platform is a valuable asset, especially when aiming at an agent that can combine the extended flexibility, robustness and capability to autonomously acquire the knowledge with more fine-tuned cognitive capabilities and the accuracy that can be achieved when modelling very specific restricted domains. In the RASCALLI system, the different approaches are still separated and further research into possibilities for integration is necessary.

Integration of tools

Since the tools in the RASCALLI platform are isolated modules, they can be separately evaluated, according to what information they have gathered, what outputs they have produced to certain tasks or according to the user satisfaction. Consequently, different realizations of Rascalli can be assembled and tested to find an optimal set of system components for a given objective and a specific usage scenario. (Automatic) integration of individual tools with the Butler's knowledge, however, clearly needs further research.

Privacy and Security

If a Virtual Butler system is to get anywhere near the example described in "Tina and her butler", several critical security and privacy issues need to be dealt with. Such a system collects a variety of information on the user over a considerably long period of time. This is highly sensible. On the one hand, data ideally should only be used with full consent of the user. On the other hand, it is unrealistic that an average user will be able to fully control the system. Just think of today's Windows operation systems. Thus a new design for the interaction between system and user is required where the system makes accessible crucial processing steps to the user, i.e. what the system is doing within a given context, why the system is doing it, and how the system behaviour can be changed. Moreover, a clear separation of privileges is required. There are data that may be given out unconditioned, others may only be given to partners or services trusted by the human user. From a today's perspective, it is highly unlikely that an ECA interface or a humanoid robot will do the job, given the problems we still face in robustly processing the input signals, in interpreting the context, and in modelling human communicative behaviour.

Obviously, a Virtual Butler will have a number of functionalities that are derived from or inspired by human cognition such as being able to act autonomously, to set appropriate actions given the current context of action and the given user preferences and interests, to find alternative task solving strategies. The user must be able to follow, control and correct the decisions of the Butler. In other words, all actions of the Butler need to be explainable and traceable. Additionally, there needs to be a mechanism that enables the Butler to inhibit certain behaviours that were not approved of by the user. This requirement clearly limits the kinds of learning approaches that can be used in connection with critical data.

One of the most important characteristics of a human butler, accompanying someone on a day-to-day basis, is data security and discretion. Even though a complex task, for the realization of the former well established technologies are available. As regards the latter, the context needs to be sufficiently modelled. The main problem that an artificial butler will encounter is that it will not be able to develop a notion of embarrassment or trust in the way a human does. There may be contexts where seemingly unproblematic data may lead to situations embarrassing to the human. To be able to autonomously decide when to use what data for what purpose, and also when to take a certain action in general, a Virtual Butler would require an understanding of the context of situations. Here again, the research community still has a long way to go.

Acknowledgments

This research is supported in part by the EC Cognitive Systems Project IST-027596-2004 RASCALLI (Jan. 2006 – Dec. 2008) and a grant by the Federal Ministry of Economics and Labour of the Republic of Austria. We would also like to acknowledge the work our partners in the RASCALLI project, in the context of this paper in particular Radon Labs for developing and implementing the 3D ECA interface, and the Research Studio Austria - Smart Agent Technologies for developing and implementing the Web interface, as well as NBU for their work on adapting the DUAL cognitive model to the Rascalli system.

6. Literature

[Irran et al, 2006] J. Irran. Kintzler, P. Pölz , , Grounding Affordances. In proceedings: Trappl R. (ed.): Cybernetics and Systems 2006, Vienna: Austrian Society for Cybernetic Studies, ISBN 3 85206 172 5, (2006).

[Irran et al, 2007] J. Irran, G. Sieber, M. Skowron, B. Krenn, Acquisition and Exchange of Knowledge From Real to Virtual Embodiment, In Proceedings of the Symposium on Language and Robots, (2007).

[Jaynes, 1957] E.T. Jaynes, Information Theory and Statistical Mechanics, in Physical Review May 1957 Volume 106, No. 4 (pp. 620-630), (1957).

[Kostadinov, Grinberg, 2008] S. Kostadinov, M. Grinberg, The Embodiment of a DUAL/AMBR Based Cognitive Model in the RASCALLI Multi-Agent Platform. Proceedings of 8th International Conference on Intelligent Virtual Agents, pp. 356-363, (2008).

[Krenn, Schollum, 2008] B. Krenn and C. Schollum, The RASCALLI Platform for a Flexible and Distributed Development of Virtual Systems Augmented with Cognition. In Proceedings of the International Conference on Cognitive Systems (CogSys 2008), (2008).

[Skowron et al., 2008] M. Skowron, J. Irran, B. Krenn, Computational Framework for and the Realization of Cognitive Agents Providing Intelligent Assistance Capabilities, European Conference on Artificial Intelligence (ECAI2008), The 6th International Cognitive Robotics Workshop, pp 88–96, (2008).

[Pfeifer, Scheier, 1999] R. Pfeifer, Ch. Scheier, Understanding intelligence. MIT Press, Boston, MA, USA, 1999.