Modelling Personal Service Assistants with Personality: From Metaphor to Implementation

Yasmine Arafa, Patricia Charlton, Abe Mamdani
Imperial College of Science, Technology & Medicine, UK
{y.arafa,p.charlton,a.mamdani}@ic.ac.uk

Abstract

In this paper we examine the Personal Service Assistant (PSA) metaphor and analyse the requirements to support public information and service access. The PSA is still in the evaluation stage as researchers develop an understanding of the metaphor and the process for realising it. An emerging dimension to realising this metaphor is the desire to physically visualise the PSA in a life-like figure that not only exhibits intelligence in the manner it reacts to human interaction, but further portrays itself as an individual with character.

We discuss the experience gained from using the PSA, providing both a metaphor and infrastructure requirements as well as an evaluation of development. This paper concentrates on a base-system infrastructure for supporting PSA development, and considers how this prototype methodology can be expanded to support a richer set of social features that would lay the foundation for developing PSAs with life-like qualities.

Introduction

The implementation, we describe here, was to support the Kimsac (Kiosk-based Integrated Multimedia Service Access for Citizens - supported by EU's ACTS 030 programme) project. The project aims to provide a flexible way for citizens to access intricate and complex information provided on public information kiosks. To provide a high degree of usability, the concept of a Personal Service Assistant is used, and is realised through adding the agent dimension to a multimedia presentation. We discuss the experience gained from using the PSA, providing both a metaphor and infrastructure requirements and evaluation of development.

This paper concentrates on a base-system infrastructure for supporting PSA development, and considers how this prototype methodology can be expanded to support a richer set of computationally feasible social features that would lay the foundation for developing PSAs with life-like qualities. Providing a foundation upon which we may bridge the gap between current research in psychological theory and practical implementation of social multi-agent systems. We are viewing the use of personality, here, from a technology view with the hope of gaining a usability advantage point. We demonstrate that a personality-based system needs the following as a minimum:

- General agent components as needed in multi-agent systems
- Social model upon which agent emotions and hence personality are derived
- Visual framework for providing an interface, interaction, information delivery, character manifestation
- An interface between the visual framework and multi-agent systems

Our solution to this third requirement was to design an asset description interface and build a tool to support such requirements. We are looking at embedding emotional indicators in these assets passed as the content of the agent communication language and investigating the criteria and ways this may vary according to the context of the conversational acts and using this as a base for building agent's social and adaptive behaviour.

The paper is organised as follows: overview of metaphor to implementation, central tool for supporting agent mediated multimedia systems for core infrastructure for the PSA and extensions to support the more interesting aspects of personality in the interface. The final section gives some concluding remarks.

An Overview of the PSA metaphor and the core components implemented

The PSA metaphor encapsulates two contrasting requirements: service access and user access. The purpose of the metaphor is to aid the design of a user interface and construct the support for this user interface for the provision of services. There are many levels to a user interface design which need to be managed, some of these design issues may require an understanding of the types of services that are to be made available, technology to be used and the users.

To summarise, designing a system to support a PSA metaphor, which provides an extensible architecture, requires that the PSA component be loosely coupled to the service providing systems. Multi-agent technology aims to provide loosely coupled systems to enable a system to be extensible through a common communication protocol. If an implementation of a PSA is to operate in an agent technology then it will require as a minimum the core components that any agent system requires. However, the PSA also needs to be able to support the user in using the system and therefore needs a computational model of the system interaction mechanism (given that natural language voice recognition systems do not reliably exist yet). A rich method for interaction and information visualisation is multimedia.

Having understood the general requirements of the PSA and the limitations through the application of the PSA the designing of the PSA at a more concrete level needs to be examined. As an intuitive metaphor the PSA offers a set of roles that define the behaviour of the metaphor. The design concerns are summarised as follows:

- Separation of the visual requirements and the reasoning requirements of the system into separate components
- PSA is a point of reference for advertising new services
- The components of the PSA as agents themselves
- Content language for sharing of information between services and the user, co-ordinated or for service access.
- The visualisation of service information
- The visualisation of the Personal Service Assistant.

As a result of these requirements, the KIMSAC architecture defines five major parts to developing a PSA architecture. Each part plays an important role in providing infrastructure and integration of numerous subcomponents.

1. Intelligent Presentation System: Multimedia content rendering and management (screen layout)
2. Multi-agent architecture for high level reasoning (service agents, agents supporting PSA functionality)
3. Distributed service management platform for heterogeneous systems (across platforms and languages technologies) and general service access to legacy databases, security and system management.
4. Distributed ontological database to support multimedia content re-use and agent access to multimedia content from a task view.
5. Rich communication language to share information and service access at a flexible level, thus permitting both sub-components and major components to be integrated in a loosely coupled manner [Charlton *et al*, 1997].

The first two components are normally understood as requirements for building a PSA architecture. The design, implementation and integration will depend on whether the visual requirements are separated out [Charlton *et al*, 1997 & 1997a]. The third component may bear some influence on the design of the architecture if there is a clear requirement for connecting to other platforms (e.g. access to databases). The fourth component is the least developed area for supporting the PSA architecture. However, it is this part of the system, which will enable the many components of a PSA architecture to be integrated in an extensible way. It defines two levels: content re-use (multimedia content is made domain independent) and agent access to content from a semantic task oriented view. This latter component is usually supported in systems in an implicit way. The support is usually provided by defining implicit interfaces between the parts of the system. The interfaces at a multimedia content level will often be hard-coded and shared between systems in an ad hoc manner. There are now some tools available that can help define some general interfaces [Tools], but what is lacking is an end to end approach that supports each major component development and an understanding of how these components can be put together at a content/context level. Kimsac developed Asset descriptions to support "high level" integration, it is this aspect of the Kimsac system which will be described to explain how the system can be expanded to include other aspects of personality in the interface.

The fifth design component in the KIMSAC architecture is essential to high level computational systems interacting - communication language. A communication language is itself understood in terms of a number of layers, protocol, syntax, semantics and channel. Part of the syntax and semantics must support the ontological level defined to integrate components in a loose, flexible but consistent manner.

Figure 1 outlines the core multifaceted requirements of a PSA type architecture. The asset tool supports structure for development and interfacing components. The interface design provides the structure and visual requirements of the metaphor. The metaphor was not easy to represent. However, the metaphor has been invented for interface development to support new paradigms of computational interaction and as such needs new tools to support representation. This latter issue will become less of a problem as technologies are developed. Extensibility, which really emphasises our core system, is supported by the multi-agent architecture, the Agent Service Layer infrastructure, multimedia content decomposition, common ontology, general interface modelling through the Asset Tool and flexible visualisation of multimedia through a presentation system. Kimsac's final system uses both the PSA metaphor and direct manipulation to support the requirements of the user. The current results from usability studies are encouraging. However, more information should be available later this year.

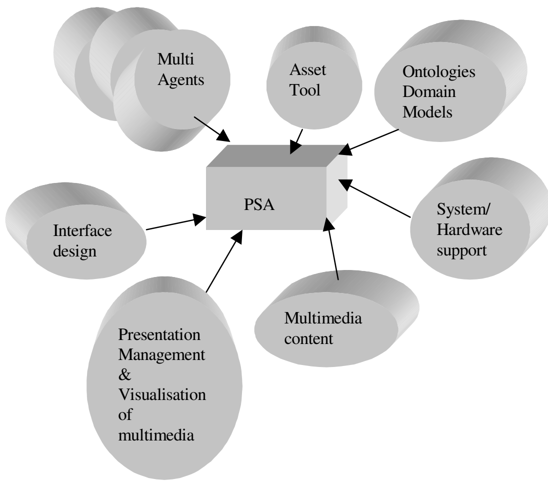

Figure 1: *Multifaceted requirements of the PSA type architecture*

Further to these core requirements for a PSA mediated infrastructure, is the growing interest of anthropomorphizing the PSA visual aspect. Much of the current research is focusing on mechanisms for effective, life-like representations of an animated PSA figure that not only exhibits intelligence but also has a character. To manifest such life-like character two technologies are adopted: multi-agents with capabilities to understand and reason about human-like qualities and multimedia. Assets, as discussed in the following section, are what facilitate the conversance of these technologies. The next section further describes the considerations for adding a personality-based dimension to the PSA and how asset descriptions can contribute to providing the required semantic understanding of the surrounding environment as well as the content being handled in the conversational acts.

The asset description and extension for a Personality Model

Objects used in the Kimsac system are termed *Assets*, which are system entities that are made up of services, data or renderable screen objects. These assets are intended for the exchange of knowledge during agent communication as well as to facilitate for content reuse. The asset descriptions provide a semantic view of the content being handled giving the agents insight knowledge of the conversational acts and thus allowing agents to reason about and react accordingly in a more intelligent manner. Multimedia content is viewed in terms of service-provisioned descriptions that provide visual meaning for the presentation system as well as augmented attributes instantiated when used within a particular application domain. The contextual semantic, visualisation entities, as well as the relationships among objects are what explicitly model the assets. The assets are abstracted encapsulations of this information providing a means by which an agent can interpret the underlying semantics of the object being used and by a presentation system to understand the mechanisms and services required for rendering the multimedia objects.

Object type	specifying the multimedia type, service,	
Content-based	describing the object content which will contribute to its functional semantics,	
Non-content-based	including conceptual and physical properties. Conceptual properties include:	
	domain	subset of the ontology
	tasks	denoting actions or functional roles
	scripts	to represent a sequence of instructions
	Physical	***properties***, such as layout and location, denote attributes that aid in the physical representation of visual objects

Asset Description Structure

The underlying conceptual model of these assets is that they are object abstractions that provide content-based data which, translates into knowledge about the multimedia content. The asset abstractions are described as meta-data templates that encapsulate semantic annotations of the Media Content, Domain (service provided), Tasks and Action models. The meta-data template is defined by a meta-data language. The structure is the ontological level between the meta-data template and the meta-data language and is necessary if the language definitions need to be changed (common in computationally reflective systems). This is the level at which asset descriptions and service descriptions have a common model.

The actual structure for an asset template takes the following set of terms: header, content (composite or atomic), domain, script and task. Each term is specialised by a set of phrases, which are applicable within a service model. We have developed syntax and a structure for describing Assets. The description language-base includes a terminology and a set of basic atomic properties that are defined as contextual implications giving the described objects their functional semantics. An Asset tool was developed to provide an interface for describing and maintaining assets. It is a general-purpose mechanism that associates descriptive properties to the defined assets.

The multimedia data model defined here, addresses the aspects related to data presentation, manipulation and content-based retrieval. It provides a set of requirements based on which the asset manager can function. The asset model is to contain many asset descriptions, which will allow the selection of assets for potentially any purpose. The purpose of this model is to dynamically generate objects for the Presentation System to render or for use with other applications that require objects of similar characteristics. The importance of the flexibility of the model is to not only to facilitate asset reuse within one application but moreover to allow other applications to also reuse these dynamically generated assets. The base Asset model consists of:

- Multimedia Description, providing an annotated view of the raw data;
- Visual Description, aiding a presentation service in describing the temporal and spatial relationships among different assets;
- Content Semantics, giving a semantic view of the stored multimedia objects; in addition to
- Service Descriptions, instantiated according to an application domain

It is worth noting that it is the script part of the asset description offers flexibility by embedding the character of the agent. The agent can define how a form of communication can be understood in this part of the description and also defining how the returning communication should ideally be composed. The form can be used to express the agent's attitude – this is optional and requires that the receiving agent share the same understanding of attitude. However, the actual composition of a message is left to the sending agent

Adding Emotional attributes as a basis for defining a PSA's personality

By including asset descriptors in the content part of the communication language between co-operating agents we provide an added dimension to agent comprehension of the actual meaning within the content exchanged in a communicative act. The assets not only serve as data that agents manipulate but can further serve as a basis for defining a social structure. Since any information can contain vast amounts of *embedded* knowledge, asset descriptions can therefore have potential to convey associated and inferred perception of the environment. Adding social attributes that can collectively serve as choices that influence a PSA's perception about the underlying semantics of the content which can attribute to it's behavioural and personality traits, provides an improvisational basis that would serve as a key aspect in determining a PSA's character.

However, to define the attributes that are required for building individual personality traits we must first define what is meant by personality (for a more detailed discussion on personality refer to [Hayes-Roth *et al*, 1997]). The concept of personality is yet not well defined, especially in a computational sense, nor is there a common view of what attributes will contribute to an individual's character. Many views extend on this issue to consider characteristic action and reaction patterns, how the outside world is viewed, and a mediation of either or both of these. All suggest that personality is individually distinctive and is to some degree independent of an immediate environmental setting, and that personality is maintained over time yet subject to some modification by ongoing experience [Walker *et al.*, 1997]. Most often, behavioural reactions are greatly influenced by current mental state, and innate emotional experience; therefore, we consider personality to be closely dependent on emotion, despite the apparent differences between the two. As personality is seen in terms of temporal consistency that is usually stable over a life-time, while emotion, on the other hand, is seen as the exact opposite, as temporal inconsistency which is usually short-termed and greatly influenced by the immediate situation [Allport, 1961]. Despite these differences they, nonetheless, overlap in that it is the emotions that influence behavioural tendencies hence develop personality traits.

Personality Model

We assume that a character need not exhibit all the attributes of the personality definition to be a successful personality model. It needs to incorporate at least some of the more basic features of personality. We try to model the broad qualities of personality that include individually distinctive, and consistently enduring yet subject to influence and change. To realise a Personality model, we define a finite set of Personality Types and Manifestation Styles that collectively serve to define the overall representation strategy by which the individualised behaviour a chosen PSA character can be visualised. These manifestations will define the various modalities in which personality may be presented. For simplicity, we concentrate on the visual (motional and gesture) manifestations, but can later extend this to include other modalities like synthesised speech.

Personality types:

Active (motional)
Quiet
Reserved ….

Manifestation styles:

Motion: Hand	Gesture: Facial
Body	Hand
Arms	Body
Legs	Combined
Combined	
Combination	
Other modalities	

The emotion(s) that results from individual situations will influence how the representation strategy will be defined and hence generating personality traits based on the presupposed personality type. In order to support dynamic creation of representation strategies for visual character-based manifestations there needs to be a semantic understanding of the content being handled in accordance with the context within which it is being conveyed. We extend the asset descriptions to include Provisional Indicators to an expected emotion that a given content may convey. We propose, to start with, a limited number of basic (fundamental) emotions that can be anticipated when handling a particular asset. These indicators are included in the non-content-based descriptor part of an asset description.

Basic emotional indicators:

Negative	Neutral
Positive	Ambivalent
…..	

Attitude:

Friendly	Combination
Cautious	
Passive	
Aggressive	
Polite	

These four indicators alone can not determine behavioural reactions. What may have a negative impact on one individual may have neutral or yet positive on another. This impact would depend on past experiences (in general, as well as, experiences with particular individuals), the current mental state and the environment. So the actual influence of such indicators will be determined by these variable factors that are governed by the context in which the content is being conveyed. Nevertheless, these indicators serve as the innate knowledge an agent may start with, and based on which may influence its behaviour. If we are effectively able to use such variables in defining the context we may effectively determine the attitude form and visual realisation of a PSA's character.

Therefore, a further requirement to the PSA's perception is not only an understanding of content but also and understanding of the context and how this context may influence the behavioural attitude. The variable factors that contribute to the context in which a communication is being maintained are the PSA internal states and the immediate environment, examples of which are as follows:

Immediate environment:
System resources
Access time
Availability

Agent/User:

Response time/delay	Mental State
No. of requests	Social Attitude
Importance	
Immediate memory	
Past experience	

Variable factors influencing the immediate situation

Since the semantics associated with the data in an asset will typically change over time depending on these variables, it is important to dynamically change the set of emotional tendencies associated when defining a representation strategy. A further variable factor can define the degree or intensity of the emotional reaction. This can be determined from the same variables described above to define the extent of emotion expressed when defining the representation strategy. Figure 2 illustrates the basic components of a PSA personality model. To incorporate a personality model with the PAS architecture we define three modules based on which a PSA can mediate behavioural responses.

Social interaction realised through the conversational acts between agents and the PSA further affect behavioural responses. The communicative intentions can form a basis on which the PSA may infer social and behavioural attitude. This may be structured by associating intentional inference to the constructs when its constraints are satisfied. The PSA must maintain its memory of past dealing with agents and their social behaviour and attitudes towards him. This memory serves to influence or constrain future social interactions with familiar agents, allowing the PSA to alter its own behaviour in a manner in accordance with the other agent. Changes in attitude and resulting social emotion will influence the representation strategies appropriate to interpret this emotional behaviour in accordance with its innate personality.

We have presented the considerations required for defining a Personality model based on embedded emotion that is further influenced by the immediate environment. However, due to space limitations details of the represent-ation strategies with which the behavioural and personality visual manifestations have not been included here, simple representations are in [Mc Guigan *et al.*, 1998].

Related Work

Much of the recent work into visual agent characters and personality representation is directed towards creating animated behavioural agents and synthetic actors for entertainment systems. Although there exists various commercial attempts to visualise the PSA for example the Microsoft Bob [Moffat, 1997], the Parrot Peedy [Microsoft, 1995], most are constrained by repeated mimics of pre-scripted animated actions. The Oz projects does, however, define an architecture that integrates mechanisms for perception, emotional behaviour and some social knowledge, the behavioural responses and corresponding representations are not dynamically generated but rather determined by a predefined script. The main criticism to the Tok architecture used to animate Woggles characters in the Edge of Intention scenario [Ball *et al.*, 1996], that is realised within the framework of the Oz project is that it explicitly models behaviour and is limited in the dynamic creation according to situation.

Although, such agents do exhibit a certain degree of intelligence this is in a sense lost amidst the mimicked repeated behaviour that misleads the impression a user takes. This could have negative affect on the human interact for dismissing most of the reactions an agent may come up with for the mere reason of lack of sophistication

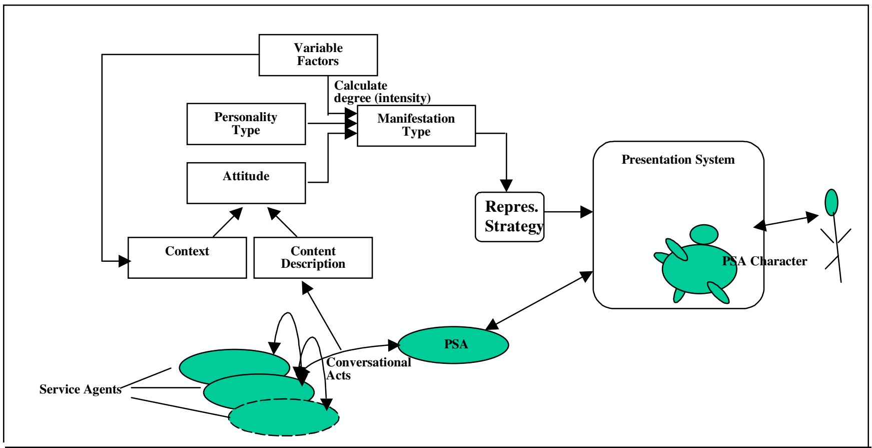

Figure 2: *Personality Model*

and exhibition of intelligence which results in the wide dislike to such an interface. Another example is the work on Linguistic Style Improvisation, [Reilly & Bates, 1993], has realised agent personality through verbal speech modality. They have developed mechanisms that are able to make generic choices about linguistic style that is closely related to human perception of personality and provide a source for linguistic behaviours for life-like computer characters.

Conclusion

In this paper we have approached the design and development of personality-based systems from a technology requirement. The demands of the Kimsac project illustrated the need for using the PSA metaphor. The metaphor itself raises the questions from a technology perspective whether interfaces to computer systems is better supported through life-like manifestations as opposed to using more traditional methods. It is almost a natural step from considering the animated character to investigating whether personality representation can help in both generalising the technical requirements for animation and assisting in personalising the information in a more intuitive way for the user.

Our motivation for building agents systems is to solve real world problems as illustrated by the Kimsac project. We are viewing the use of personality from a technology view with the hope of gaining a usability advantage point. However, it is worth noting that building multi-agent systems is a none trivial problem and is still in its early stages of development. Also, although the PSA metaphor is itself a very attractive metaphor for building flexibility in both the interface and the service delivery, it is none the less a complex model to develop and support. The PSA requires all the functionality of a multi-agent system plus the visual interface. In today's technology the visual interface will in many systems be a multimedia interface. There are complexities in integrating multi-agent systems and multimedia. To solve this aspect of the Kimsac system its was necessary to design a number of interfaces and protocols that provided certain degrees of flexibility required when providing a PSA type interface. The result of these design requirements was the asset description, which provides the essential interface between multimedia content and multi-agent systems. To achieve personality-based aspects as a fundamental part of a software infrastructure means that the discussed designs and development are essential. Although it is possible to build these systems in a more standalone type fashion this will limit the extensibility of such systems and will mean that they will have only minimal use as they will lack in re-usability in terms of development of real world solutions. The trade off is always one of simplicity over flexibility. The extension to the original asset description was to demonstrate how to represent visual content in a consistent and re-useable way to automate the management of personality-based systems.

References

Allport, G. W. 1961. Pattern and growth in personality. New York: Rinehart & Winston.

Ball et. al., 1996. Lifelike Computer Characters: The Persona Project. Microsoft Research. http://www.research.microwsoft.com/research/ui/persona/home.html

Bradshaw, J. M.; Dutfield, S.; Benoit, P.; and Woolley, J. 1997. *KaoS: Toward An Industrial-Strength Open Agent Architecture*. Software Agents: AAAI Press/MIT Press 375-418.

Charlton, P.; Mamdani, E.; Olsson, O.; Pitt, J.; Somers, F. and Waern, A. 1997. An Open Agent Architecture Supporting Multimedia Services on Public Information Kiosks. In *Proceedings of the Second International Conference on Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM '97)*, 445-465. London, UK: PAC.

Charlton, P.; Mamdani, E.; Olsson, O.; Pitt, J.; Somers, F.; and Waern, A. 1997a. Using an Asset Model for integration of Agents and Multimedia to Provide an Open Service Architecture, Multimedia Applications, Services and Techniques. In *Proceedings ECMAST'97, Lecture Notes in Computer Science vol.*1242:635–650.

Hayes-Roth, B.; van Gent, R.; and Huber, D. 1997. *Acting in Character*. Creating Personalities for Synthetic Actors: Springer.

Moffat, D. 1997. *Personality Parameters and Programmes* Creating Personalities for Synthetic Actors: Springer 120-125

Rdemond, WA. 1995. *Microsoft Bob*, Microsoft.

Reilly, W.S., and Bates, J. 1993. Emotions as part of a broad Agent Architecture, *Working notes of the Workshop on Architectures Underlying Motivation and Emotion*.

Mc Guigan, R., Delorme, P., Grimson, J., Charlton, P., Arafa, Y. 1998. *The Reuse of Multimedia Objects by Software in the Kimsac System*. Forthcoming *in* OOIS'98.

Tools: http://www.agentbuilder.com/, http://www.bitpix.com/business/main/bitpix.htm, http://www.bitpix.com/business/main/iaf/iafintro/iafintro , http://www.networking.ibm.com/iag/iagsoft.htm

Walker, M. A.; Cahn, J. E.; and Whittaker S. J.. 1997. Improvising Linguistic Style: Social and Affective Bases for Agent Personality, In *Proceedings of ACM Agents '97 Conference*, 215 –224. ACM.