Interactive Product Browsing and Configuration using Remote Augmented Reality Sales Services

Ross Brown[1], Hye-Young Paik[2], and Alistair Barros[1]

[1] Science and Engineering Faculty, Queensland University of Technology
[2] School of Computer Science & Engineering, University of New South Wales
{r.brown, alistair.barros}@qut.edu.au, hpaik@cse.unsw.edu.au

Abstract. Real-time remote sales assistance is an underdeveloped component of online sales services. Solutions involving web page text chat, telephony and video support prove problematic when seeking to remotely guide customers in their sales processes, especially with configurations of physically complex artefacts. Recently, there has been great interest in the application of virtual worlds and augmented reality to create synthetic environments for remote sales of physical artefacts. However, there is a lack of analysis and development of appropriate software services to support these processes. We extend our previous work with the detailed design of configuration context services to support the management of an interactive sales session using augmented reality. We detail the context and configuration services required, presenting a novel data service streaming configuration information to the vendor for business analytics. We expect that a fully implemented configuration management service, based on our design, will improve the remote sales experience for both customers and vendors alike via analysis of the streamed information.

Keywords: sales product configuration, sales context management, augmented reality sales services, virtual world sales services

1 Introduction

The explosion of online retail Web sites has brought with it much convenience for consumers of goods [1]. However, some items remain problematic to sell remotely, requiring physical interaction in order to be examined effectively.

Typical web collaboration tools, such as audio and video environments [2], do not sufficiently provide an insight into the actual context of actions by the remotely connected sales assistant. The customer may struggle with understanding the actual artefact that is being discussed by the sales assistant, and a lack of relevant gestures and spatial organisation makes the process of service provision by sales staff problematic.

We highlight two main problems with conventional online sales assistance provision, the first is the lack of an inherently spatial representation of the product and services, the second is the lack of ability for remote sales staff to adequately interact with customers in order to provide assistance with the product and other aligned services.

Augmented Reality (AR) represents a view of a physical (real-world) environment whose elements are augmented by computerised sensory input and is often used to enrich user experience and perception of the real world.

Recently, the emergence of cloud computing and its large scale data processing capabilities have contributed to the renewed discussion of AR applications. Using ubiquitous and powerful cloud platforms as a data storage and process backend, users are able to experience fully immersive AR environments, even through mobile and wearable devices with relatively small computing power (e.g., Google's Glass project).

In this work, we aim to create more engaging and realistic virtual shopping environments using AR, in order to allow the customer to experience, interactively, convincing representations of physical products in their own home using a mobile device. In our platform the products are juxtaposed with their home environment in order to assist goods browsing and selection.

Extending an earlier proposal introduced in [4], we introduce the following new contributions:

- We represent customer interaction actions as various edit actions on the scene representation, and propose a version management mechanism to support navigating the different versions of product configurations to help the customer with the analysis tasks.
- We propose an event streaming model which sends customer interaction events (e.g., touching an object) to the vendor for management purposes.

We present the framework, the key services offered, interactions within a typical session and illustrate with a use case of a large screen television sale using such a framework. We then conclude with discussion of how this preliminary work may be further developed.

2 AR Sales Service Framework Overview

2.1 A Motivating Scenario

Our example customer wants to purchase a large screen TV that will be compatible with his own audio system and the furniture at home - in terms of the size, appearance, technical specifications, etc. The customer is at home and places an augmented reality registration pattern onto the ground, where the actual television is to be installed. A dialog appears requesting customer information to create various contexts - e.g., customer location, size of the room. Then, a sales representative is alerted to the creation of a session, and logs in from the other end to assist with any questions. The user is also able to "load" into the space some of his existing products such as the audio system and entertainment unit, using our 3D Content Service. The user requests to view large screen televisions from a specified brand. He now can explore the product configuration by performing actions such as placing the TV onto his existing entertainment unit, re-locating the furniture (along with the selected TV) within the room, trying

out different colours, probing into technical specifications and compatibility issues with his audio system. The system allows him to "track/navigate" through the history of interactions, enabling him to go back and forth to any point in time and examine the setting. The system also understands the complex contexts within the objects in the interaction space such as 'TV set placed on an entertainment unit' and how to associate transaction and recommendation services with those objects. Figure 1 shows our prototype AR sales service architecture, with a centre image of a large screen television, entertainment unit, stereo and avatar of remotely connected sales assistant demonstrating the entertainment system to the client.

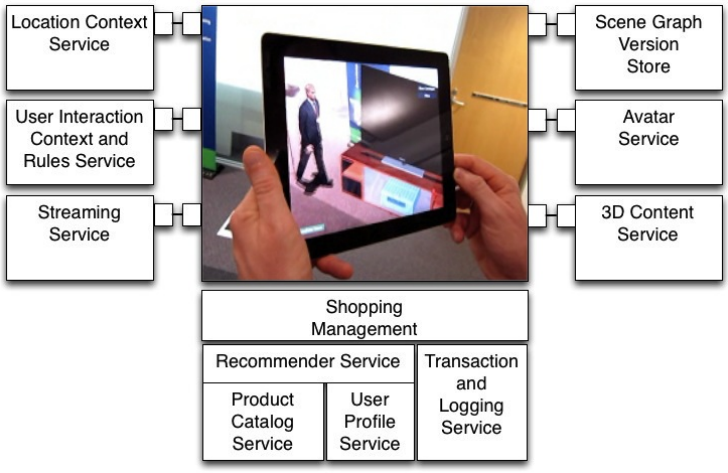

Fig. 1. AR sales prototype service overview, image of running prototype in the middle

2.2 Preliminaries

Services The AR sales service framework (Figure 1) is supported by four categories of high-level services: Scene Graph Management and Rendering, Content Management, Context Management and Shopping and Service Recommendation Management. The focus of this paper is on the Context Management services. Here we briefly explain some of the other service components the framework relies on.

The Avatar Service (in Content Management) provides a graphical representation of a sales assistant (which can be a chat bot or human controlled avatar). It is also possible that any friends of the customer who are connected through a social network can join to collaborate during the sales sessions. For simplicity, in this paper we assume that the main actors are the customer and sales assistant.

The Product Catalog Service (in Shopping Management) interfaces with the product catalog information available in the system. They provide comprehensive

information abouta product, including, amongst other things, all configuration options, recommendations, related product suggestions/alternatives and stock level.

The 3D Content Service is responsible for communicating with the product service and 3D rendering engine to send 3D representations of products into running sessions with a customer. The service also understand the compatibility information (in terms of dimensions and weight, for example) with other available or adjacent products.

Representing The Customer Location: Marker-Based Approach One of the core concepts of our approach is to be able to represent the customers current location and surroundings (in a house) in a virtual environment [4]. Doing so involves two kinds of location-related information which is managed by the Location Context Management Service: *absolute* and *relative* coordinates. The absolute coordinates are derived from the customer's physical location. In our application scenario, absolute coordinates can be the home address of the customer. The relative coordinates are based on an adaptive spatial subdivision of the house, known as a Regions. This structuring is derived from the observation that the internals of the home of a customer are divided up into rooms and form a natural context for purchases within a house.

3 Context Management in AR Sales Service Framework

In this section, we explain the context in our system and how it is modelled and managed in scene product configurations. Broadly, there are three types of context in our system, depending on where and how we obtain the data.

- Static and Spatial facts: location of the markers used, dimensions of the room, physical size of the objects, etc.
- Profiled facts: user account details, preferences and profile (gender, age, job, etc.)
- Derived facts: the information generated by user interactions with the scene. This context can be used also to capture/extract user's interests by events like camera movements, etc.

We have conceptualised the context management of the scene via a versioning tree structure which manages the editing contexts and interacts with the scene graph management system to render the final product configuration to the mobile viewer. Scene graphs are commonly used in graphics applications [5], and are designed to be an efficient data structure for editing and rendering of 3D scene databases in graphics applications. We use the Model-View-Controller pattern to manage the interactions between the user interface and the underlying product representation structure (i.e., the scene graph). With the controller component being a context management service inside the software responding to customer edit actions, contributing to an audit trail of customer interactions with products.

3.1 Customer Actions and Scene Graphs

We identify the following customer interaction actions as the source of generated context in product configurations:

- Interacting with a single object in the scene. These actions trigger editing of product representation node attributes.
  - Addition/Modification/Deletion of object(s) - instantiations of new objects (and combinations thereof) in the scene.
  - Modification of object material(s) - significant changes in object attributes (e.g., material properties)
  - Modification of object relationships - moving an object in the scene, as users only have a translation and maybe a flip function available as objects have a typical orientation
- Aggregated object actions. This type of interaction causes not only editing of scene representation nodes, but also propagation of scengraph changes throughout all nodes according to thematic rules.
  - Application and changes in the themes of the scene (e.g., colour scheme). The change propegates across a number of objects in the scene.
  - Making large changes to the objects, based on relative organisation (e.g., objects that are 'spatially related'). Similar concepts to copying and pasting multiple objects in the scene.
  - Granularity of attribute locking - eg. one object is fixed as others change.
- User Specified Contexts - explicit creation of contexts led by the user. Representing a parallel list of product configurations, and context change sets, that are assumed to be independent, but can be copied and pasted to different context sets.
- Camera View Changes - context changes when in different viewpoint are sent to the sales person as useful sales and usage information, eg. how is the object being used in the house, and its physical context.
- Backtracking with locks on previous changes, preventing propagation.

3.2 Context Versioning Fundamentals

Each of the editing operations is stored as contexts within a context versioning structure [6]. This versioning structure is then used to provide configuration management for the user purchases. This is similar to other versioning systems, with the model being based on document structures for the product information. We conjecture that a versioning tree of these atomic commands represents the overall context management for a configuration of products over the time of the sales session. Modifying a tree modelling approach from [6] and [7], we describe the configuration versioning performed in this retail application. The intention is to show that the scene graph can be reconstructed from a versioning tree, to show a purchase configuration at event E_i, and to show that the data structures and interactions are capable of meeting all context management requirements for users as shown above. The structure of the scene graph is controlled by the following commands, which map to user actions on the purchase configuration:

- $C_{INS} = (N_{ID})$ - node ID to insert into the scene graph from a 3D content service site similar to Google Warehouse [8].
- $C_{DEL} = (N_{ID})$ - node ID to remove from the scene graph.
- $C_{MAT} = (N_{ID}, P_{ID})$ - node ID and product info ID to change in the Scene Graph, ie. modify with new information.
- $C_{TRANS} = (N_{ID}, M)$ - node ID and transformation matrix M (4x4 Homogenous) to load into the scene graph for N_{ID}.
- $C_{ATTACH} = (N_{ID1}, N_{ID2})$ - specifies the two nodes to attach to each other in the scene graph, ie. the objects are attached to each other, and can have group operations performed.

A *version tree* contains the above commands that may be user or system activated [7]. A traversal of the version tree generates the required product configuration version v as a scene graph for display. The requirements for this tree are the ability to add new versions of the configuration, and to then trace through the versions to generate a scene graph of the present configuration at time t for user u. Each command stored in the version tree creates a new configuration context for representation on the mobile device. The version tree commands derived to maintain these contexts are:

- *Version Refinement* - this is defined as the insertion of a command node C after the present version. Performed at the end of the version traversal trail, forming version v.
- *New Version Branch* - here the insertion is at a version v of less than V_{max} making the insertion form a new branch. This does not insert between nodes.
- *Merge Versions* - the new node is attached, and merges two specified nodes *v1* and *v2* to the new node v.
- *Present Version Return* - the version number is selected, and the tree traversed until that version number is the present node being processed and is returned as v. This is used for returning to specified versions of objects in the tree, to then progress forward with new modifications.
- *Generate Version Configuration* - the version number is selected, and the tree traversed until that version number, to generate the appearance of the configuration for version v as a scene graph.

4 Streaming Consumer Configuration Service for Vendors

In our approach, the above interactions are stored on a database at the user end. We now investigate the streaming of this context management data to a vendor for management purposes. We generate five different streams of information that can be read by the vendor:

1. Versioning - each of the above versioning tree events can be sent to the vendor for analysis, eg. addition of a new object from a catalogue.
2. Object - features that people touch and operate can be logged and processed as interaction events, eg. touching the draw on a cabinet to look inside the furniture.

3. Camera - camera movements (ie. mobile device movements with respect to the fiducial marker) are indicative of consumer interest in components of the device, eg. user may visually dwell on the controls of a sound system, to understand functionality.
4. Video - the camera stream can be processed, providing information about the setting of the interactions, and the face of the viewer, eg. the video stream can be processed to obtain the colour of the room, to find context information on user choices.
5. Audio - the audio information from the client can be streamed and processed by the vendor for future analysis, eg., vocabulary used when analysing a product.

This provides five streams of event data that can be integrated with the configuration histories, and then processed for analysis of user buying habits. An event model has been developed [9] to form a stream of event tags that can then be mined. What should be noted is the rich source of semantic media information proceeding from the AR application during its execution. Not only does the vendor and client have standard editing capabilities, which provide user interface events that are easily defined from elements of executing code, but there is the stream of more continuous real-time data that needs to be processed, including audio and video feeds, along with continuous camera orientation data emerging from the movements of the user. We detail the event groups listed in Table 1.

The first set are *Versioning* events, derived from the user interface interactions we have detailed in the use case: refinement, branch, merge and present. These are readily captured from the versionign system and deposited in the output event stream to the vendor. *Object* interaction events are recorded, which are not related to the versioning system, but involve the user touching objects of interest on the screen. These touches can also be used to highlight objects for discussion by the client and vendor.

AR systems usually use a registered location (marker or marker-less) as the centre of a coordinate system for the objects to be viewed as a 3D camera model, streaming *Camera* events[4]. This movement of the camera can be regarded as a proxy for the visual attention of the viewer [10]. The point of regard is a stream of data proceeding from the movement of the AR viewing device with respect to the registered coordinate system. This data can be streamed to the vendor to identify points of regard on the object via the camera model being used. In addition, fixations can be extracted from the stream using a visual attention algorithm [11]. The zoom, or closeness of the device to the registration point, can be treated in a similar manner, to indicate areas of interest on the objects in question.

The back facing camera on the mobile device contains a video feed of the room in which the transaction is being carried out, and is used to create the augmented view. The camera stream contains *Video* event information regarding the nature of the surrounding environment, including objects and colour schemes of the rooms. Object extraction and recognition algorithms [12] can be utilised in

these instances to provide information to the vendor regarding the visual context of the purchase. The front facing camera provides a video feed of the user to the vendor. This can be processed for facial expressions using expression recognition algorithms [13], to be indexed for analysis of the session.

Table 1. Event model derived from Customer-Vendor information stream, with associated data components in the right most column.

Stream	**Event Type**	**Components**
Versioning	Refinement	Time, VersNum, NewObjID
	Branch	Time, VersNum, NewObjID
	Merge	Time, VersNum, NewObjID
	Present	Time, VersNum
Object	Touch	Time, ObjID, InteractionID
Camera	Orient	Time, CamVec
	Fixation	TimeBegin, TimeInterv, CamVec
	Zoom	TimeBegin, TimeInterv, Scale
Video	Forward Object	Time, ObjectID, RelPos
	Forward ColorPallette	Time, ColorPallette
	Face	TimeBegin, TimeInterv, FaceExpType
Audio	Client Word	Time, KeywordID
	Client Stress	TimeBegin, TimeInterv, StressEventID
	Vendor Word	Time, KeywordID
	Vendor Stress	TimeBegin, TimeInterv, StressEventID

Audio events can be processed in a similar manner. Words can be extracted from the audio stream generating key word events, and can be readily processed with modern audio analysis algorithms [14]. In addition, approaches to audio stress detection, in a similar vein to facial expression recognition, can be applied to detect if the client is unhappy/happy with a passage of interaction with the vendor.

Such rich sales data can be streamed and stored across multiple instances of AR purchasing applications on mobile devices for up to date information to the vendor in real time, and for use later as a business analysis data source. This information can be indexed by the above event model, to identify key queries for business analysis. For example, as per the use case presented, we propose that the gaze points and interaction touches for the objects can uncover new information about how an object is perceived, and what interests people when shopping for these items using the AR system.

5 Version Tree Generation Example

We now illustrate via a brief use case how this information can be extracted from the client vendor interactions to generate a stream of information that is useful to the vendor. As the user starts up the application on their phone, the previous account, street location and region information (within the house, eg.

"lounge room") is extracted, to maintain locational context within the house of the client. We direct the reader to [4] for more details about location context management within our AR sales service framework. As the use case progresses via client interactions with the sales system, a stream of events is generated locally to maintain the sales context structures. As the version tree is created, a path through the version tree from the root to a version v is able to generate a viable product configuration. A sequence of commands from the use case is listed that generate different versions of the configuration, and in turn, different user contexts:

1. Insert Large Screen TV - user selects a large screen TV from the catalogue (Refinement Event).
2. Insert Cabinet - user selects a cabinet from the catalogue (Refinement Event).
3. Insert Amplifier - user selects an amplifier from the catalogue (Refinement Event).
4. Change Brand of Large Screen TV - user selects a new form of TV, thus a branch is formed from the base node with a new TV (Branch Event).
5. Change Colour of Cabinet to Teak - a new node is created for the cabinet location in the version tree (Branch Event).
6. Examine Amplifier - the user moves the mobile device to fixate on the amplifier controls (Camera Fixation Event).
7. Examine Amplifier Controls - the user zooms in on the amplifier by moving forward (Camera Zoom Event).

The first five events generates the following version trees, as shown in Fig. 2. If the user wishes to show, for example, version 4 of the configuration, then the tree is searched for that version number and then rendered to that point.

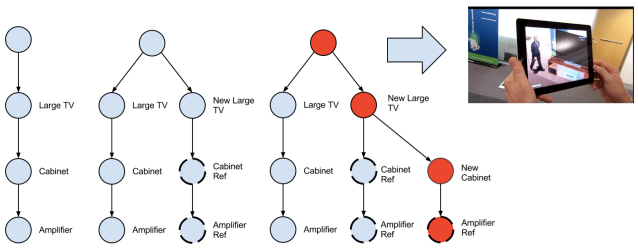

Fig. 2. Shows progression of context changes for TV purchase use case as a version tree from left to right. Note, references to nodes are shown as dashed discs, to reduce memory requirements. Darker discs in the final tree show the path taken to generate the product configuration on the mobile device

In addition to this localised retention and management of the information, other event and context information is transmitted to the vendor from the sales scenario. Events six and seven in the above list are drawn from the user camera movements via relevant algorithms, and the zoom events are extracted via similar methods.

We model our event stream as an append-only sequence of tuples, where each tuple in a stream has a timestamp (ts), event-type name and event-type specific data fields a_1 ..., a_n. The above use case generates the following stream:

- (14 August 2012 12:34:00 GMT, version_refinement, v1.0, 00987UTV)
- (14 August 2012 12:34:20 GMT, version_refinement, v2.0, 88700UYC)
- (14 August 2012 12:34:40 GMT, v3.0, 000P768UD)
- (14 August 2012 12:34:45 GMT, version_present, v1.0)
- (14 August 2012 12:34:58 GMT, version_branch, v4.0, 009888UTX)
- (14 August 2012 12:35:05 GMT, version_present, v2.0)
- (14 August 2012 12:34:58 GMT, version_branch, v5.0, 889066RR)
- (14 August 2012 12:35:01 GMT, camera_fixation, 3 sec, cam<50,10,89>)
- (14 August 2012 12:35:06 GMT, camera_zoom, 4 sec, 1.2098)

Each event type is accompanied by its own relevant data fields. For instance, version_refinement has version number and object id, whereas camera_fixation has a time interval and coordinates for the position.

6 Related Work

The retail industry is trying to overcome the limitation of conventional online shopping malls and lead innovation in future shopping experiences by adopting augmented reality technology. For example, Tesco [15] introduced a marker-based AR system where, combined with a Web cam, a customer can project a 3D image of a product onto the screen and interact with the image by moving a marker. Furniture companies like IKEA or MyDeco offer browser-based 3D planners for interior decoration [16]. Typically, these systems convert a 2D floor plan to 3D with multiple camera view angles. The backend system is connected with a large database containing product details from the manufacturers. Lu and Zhu's systems [17, 18] show an AR-based sales assistant tool where an object (such as a sofa) can be placed in a real scene and the customer can search/switch products in an AR display. Other AR systems have investigated configuration systems [19], but have not incorporated avatar representations for remote sales assistance, and have not defined streaming information to be sent to vendors for each transaction as a business intelligence source. Although these systems aim to achieve similar goals to our system, the interactions with the products are limited and do not support easy navigation of the browsing/configuration history from searching/switching products in a scene.

In terms of representing and managing context in a pervasive computing environment, almost all of the context modelling and management in pervasive computing developed so far has focused on managing various sensor data (GPS,

light, temperature, time, movement, etc.) in the physical world and architecting distinct layers to overcome the issues arising from heterogeneous sensor data sources [20]. In our application to the augmented reality environment, the issue of dealing with imperfect or heterogenous data will not be of concern as our context data is not from distributed sensors. However, being able to effectively deal with a timeline (the history of context), effective reasoning and expression of dependencies will be important.

Pitarello [21] presents work on context management for 3D environments, but only in author, and then location based within 3D environments. Cook [22] presents Learning Setting-Generalized Activity Models for Smart Spaces, in which context management is applied in homes for tracking elderly citizens using HMMs. Helal [23] worked on context management for spatial localisation within a family smart home. Thus, the context management is similar, due to the need to provide a mapping between locations and particular services. In summary, space is subdivided in homes, but is not directed by people for retail purposes. In a smart home scenario, the person is already detected by cameras and sensors to provide a spatial location within the environment. We can position this AR ability as a service within a smart house system in general as a digital overlay on the house, for physical service planning etc. - renovations with reference to other data within the environment. Thus, contexts can be maintained automatically within a smart house. For a non-smart house, we default back to explicit context management, using labels assigned to an AR marker or markerless positions.

7 Conclusion and Future Work

In this paper we have developed a data model for the back end of an Augmented Reality sales agent system we have prototyped. We have identified key context manipulation components, and have articulated the events that are generated in a stream sent to the vendor from such a sales application.

Future work involves the incorporation of a cognitive model to process the events into a semantically queryable database. From this, personalisation and parameterisation of the information stream can be performed and data and process mining can be done on the data streams to provide a complete augmented reality service information model to vendors.

References

1. Elberse, A.: Should you invest in the long tail? Harvard Business Review **86** (2008) 88–97
2. F., L., C., E., R., P.: Collaboration tools for global software engineering. IEEE Software **27** (2008) 52–55
3. Morrison, A., Oulasvirta, A., P., P., Lemmela, S., Jacucci, G., Reitmayr, G., Naanen, J., Juustila, A.: Like bees around the hive: a comparative study of a mobile augmented reality map. In: 27th International Conference on Human Factors in Computing Systems (CHI '09). (2009) 1889–1898

4. Brown, R.A., Barros, A.P.: Towards a service framework for remote sales support via augmented reality. In: International Workshop on User-Focused Service Engineering, Consumption and Aggregation (USECA 2011). (2011) in press
5. Herter, J., Schotte, W., Ovtcharova, J.: Bridging vr and item-based plm systems using an object oriented approach. In: PLM08 - 5th International Conference on Product Lifecycle Management, Seoul, South Korea. (2008)
6. Koch, C., Firmenich, B.: An approach to distributed building modeling on the basis of versions and changes. Advanced Engineering Informatics **25** (2011) 297 – 310 ¡ce:title¿Information mining and retrieval in design¡/ce:title¿.
7. Miles, J., Gray, W., Carnduff, T., Santoyridis, I., Faulconbridge, A.: Versioning and configuration management in design using cad and complex wrapped objects. Artificial Intelligence in Engineering **14** (2000) 249 – 260
8. Bustos, B., Sipiran, I.: 3D Shape Matching for Retrieval and Recognition. Springer London (2012)
9. Westermann, U., Jain, R.: Toward a common event model for multimedia applications. Multimedia, IEEE **14** (2007) 19 –29
10. Buscher, G., Dumais, S.T., Cutrell, E.: The good, the bad, and the random: an eye-tracking study of ad quality in web search. In: Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval. SIGIR '10, New York, NY, USA, ACM (2010) 42–49
11. Duchowski, A.T., Çöltekin, A.: Foveated gaze-contingent displays for peripheral lod management, 3d visualization, and stereo imaging. ACM Trans. Multimedia Comput. Commun. Appl. **3** (2007) 6:1–6:18
12. Xu, C., Hanjalic, A., Yan, S., Liu, Q., Smeaton, A.F.: Special section on object and event classification in large-scale video collections. Multimedia, IEEE Transactions on **14** (2012) 1 –2
13. Lee, J.S., Oh, C.M., Lee, C.W.: Facial expression recognition using aamicpf. Human-Computer Interaction. Interaction Techniques and Environments **6762** (2011) 268–274 10.1007/978-3-642-21605-3_30.
14. Keshet, J., Bengio, S.: Automatic Speech and Speaker Recognition: Large Margin and Kernel Methods. Wiley (2009)
15. Barnett, E.: Tesco trials augmented reality. http://www.telegraph.co.uk/technology/news/8895923/Tesco-trials-augmented-reality.html (2011)
16. MyDeco: Mydeco 3d planner. http://mydeco.com/3d-planner/ (2012)
17. Lu, Y., Smith, S.: Augmented reality e-commerce assistant system: trying while shopping. Human-Computer Interaction. Interaction Platforms and Techniques (2007) 643–652
18. Zhu, W., Owen, C.B.: Design of the PromoPad: An Automated Augmented-Reality Shopping Assistant. Journal of Organizational and End User Computing (JOEUC) **20** (2008) 41–56
19. Andel, M., Petrovski, A., Henrysson, A., Ollila, M.: Interactive collaborative scene assembly using ar on mobile phones. In Pan, Z., Cheok, A., Haller, M., Lau, R., Saito, H., Liang, R., eds.: Advances in Artificial Reality and Tele-Existence. Volume 4282 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2006) 1008–1017
20. Bettini, C., Brdiczka, O., Henricksen, K., Indulska, J., Nicklas, D., Ranganathan, A., Riboni, D.: A survey of context modelling and reasoning techniques. Pervasive and Mobile Computing **6** (2010) 161–180
21. Pittarello, F., De Faveri, A.: Semantic description of 3d environments: a proposal based on web standards. In: Proceedings of the eleventh international conference on 3D web technology. Web3D '06, New York, NY, USA, ACM (2006) 85–95
22. Cook, D.: Learning setting-generalized activity models for smart spaces. Intelligent Systems, IEEE **27** (2012) 32 –38
23. Helal, S., Mann, W., El-Zabadani, H., King, J., Kaddoura, Y., Jansen, E.: The gator tech smart house: a programmable pervasive space. Computer **38** (2005) 50 – 60