An Integrated Framework for Supporting Photo Retrieval Activities in Home Environments

Dario Teixeira[1,2], Wim Verhaegh[2], and Miguel Ferreira[2]

[1] Eindhoven University of Technology, PO Box 513, 5600 MB Eindhoven, The Netherlands
[2] Philips Research Laboratories, Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
teixeira@natlab.research.philips.com
wim.verhaegh@philips.com

Abstract. This paper addresses the content overload problem applied to photo retrieval activities in the home environment. The starting point is an analysis of the main activities that users perform with their photographs. We then discuss two different interaction paradigms, namely browsing assistants and conversational search, which can provide software assistance to help users cope with increasing numbers of photographs. The major contribution is the presentation of an integrated architecture which combines these interaction paradigms to provide the user with a home-friendly, multimodal, and seamless system for photo viewing. We also discuss the user interaction aspects, internal architecture, and algorithms of each of the two paradigms.

1 Introduction

One of the most enduring visions of ambient intelligence portraits the home as a window to the world, enabling users to access all different kinds of information in the comfort of their living rooms [1]. However, the so-called content overload problem presents a real challenge to this vision: given the overwhelming increase in the amount of available information, people must increasingly rely on software assistants to sift through all the choices they are given [2],[3]. Moreover, while significant research has been directed towards solving the problem in domains where computer-like interfaces are the norm, less attention has been paid to content overload in the context of in-home environments. As a consequence, many of the available techniques rely on interfaces which not only contradict the principles of ambient intelligence, but might even alienate some of the less technology-savvy individuals.

Most seriously, the content overload problem is not limited anymore to publicly available content such as films, music, or books. Our statement is that the introduction of new technologies, in particular the generalised use of digital cameras, has the potential to bring about the content overload issues into the domestic front as well. The main reason lies in the convenience and low-cost of digital photography, coupled with ever-growing storage capacities. As a result, we are already witnessing a significant increase in the number of photos that people take, and the consequent difficulty in managing them [4].

In this paper we describe a system designed to assist photo retrieval activities in the context of an in-home environment. The starting point is the characterisation of the various retrieval activities that people engage with their photographs, namely searching,

E. Aarts et al. (Eds.): EUSAI 2003, LNCS 2875, pp. 288–303, 2003.

wandering, and recommending. We then describe two different paradigms for accessing information—conversational search and browsing assistants—which can handle those activities. Most importantly, we present the architecture of a system which integrates them in a seamless manner. After a thorough analysis of the particularities of the photo domain, we proceed to describe in detail the user interaction aspects and algorithms used by the browsing assistant and the conversational search engine. Finally, we present some early experimental results, and discuss the following research steps.

2 User Interaction Support

In this section we aim to provide a rationale for the work described in the remainder of the paper. As a first step, we will analyse the kind of activities related to photo retrieval which are performed by users in home environments. We shall then describe the interaction means which can be mapped onto those activities in a natural way. At last, we present an illustrative use-case scenario.

2.1 User Activities with Photographs

For single-user situations we have identified three broad categories where all retrieval activities can be placed: *searching*, *wandering*, and *recommending*. Even though multi-user situations would allow for other possible activities, such as *story-telling*, in the general case these can also be mapped into the three broad categories presented (story-telling, for example, is a form of wandering through the photo collection).

The searching activity refers to the situation where the user is looking for one specific photo or a set of photos. In this case, the user has a concrete goal in mind, and the purpose of the activity is the satisfaction of that goal. The wandering activity, on the contrary, reflects no specific goal on the part of the user. It corresponds to the casual, aimless roaming through the photo collection, without any higher-level purpose other than to enjoy viewing the photos and reliving those memories. At last, we consider asking for recommendations as another possible activity for users to initiate. It differs from the previous two in the sense that rather than being an independent activity, it is more properly classified as complementary to either searching or wandering. In the former case, recommending means that the system is free to suggest some of the search parameters. In the latter, the system would present the user with suggested paths for wandering.

2.2 Interaction Means

The interaction means can be seen as tools or paradigms which support the user to perform a given activity. Our choice of interaction means was guided not only by their efficacy in aiding their designated activity, but most importantly, how well they would fit within the vision of ambient intelligence, especially in what concerns the respect for the specificities of in-home environments. In practical terms, this meant considering means which would not disrupt the normal social interactions present in the home, that could

make use of more natural modalities such as speech and objects, and finally, that could provide the users with an 'experience' and a feeling of enjoyment when using the system.

The first interaction means we shall consider, the *browser*, is similar in principle to a normal web browser. The latter displays one web page at any time; the page has connections to other pages, which in turn connect to many others, and so on. Instead of web pages, our browser deals only with photographs, and the connections to other photos are calculated by the system and presented to the user as thumbnails. This is an interaction means where the user is in control, using the photo connections as association aids to traverse the photo collection.

As depicted by Fig. 1, the browser could be used for both searching and wandering activities, even though using it to search for a photograph through a potentially huge photo collection is likely to be inefficient. However, it does lend itself naturally to the wandering activity, and this particular combination (coupled with presenting recommendations to the user) fits well into the general description of a *browsing assistant* [5]. This sort of system is designed to run continuously in the background, looking for items related to the user's current selection, and suggesting them in a non-intrusive manner. Since it may keep track of the user's preferences and browsing habits, one expects it to be able to produce very relevant suggestions. We will elaborate further on the browsing assistant in Sect. 4.

A *conversational interface*, on the other hand, is an interaction means which can only be mapped naturally onto a searching activity (again, with or without recommendation). Conversational interfaces emerged from attempts at improving classic query-based search and recommendation processes. Traditionally, query-based systems were synonymous with the so called *ranked list* approach, whose prime example can be found in the familiar web search engines. A conversational interface aims to provide better results by mimicking human dialogue interaction as the means to iteratively fill in the variables for the query [6],[7],[8]. This solution has several advantages over the ranked list approach: first of all, the user is not overwhelmed with a list of possible alternatives; instead, she can have the system play the role of an all-knowing mentor, guiding her through a series of well-thought questions, and eliminating all irrelevant items step-by-step. Moreover, it is our expectation that having a dialogue with the system will prove to be a very natural way of interaction, thus satisfying our primary criteria. To this process, where a conversational interface is used to progressively narrow down the search space until just a handful of matching items remain, we have given the name *conversational search*.

At this stage, one could point out that even though both these interaction means serve their purpose competently, we would only be replacing a computer-like interface with a two-headed system that would force the user into different 'modes' for different tasks. The answer to this problem is based on the insight that it is possible to present the user with a unified system where the transition between the different activities is as seamless as possible. The key idea is to have the browsing assistant navigate solely through the subset of photographs which match the current criteria as defined by the conversational search process.

Figure 2 presents an architecture which meets the broad requirements we have just outlined. One can see the division between the conversational search engine and the

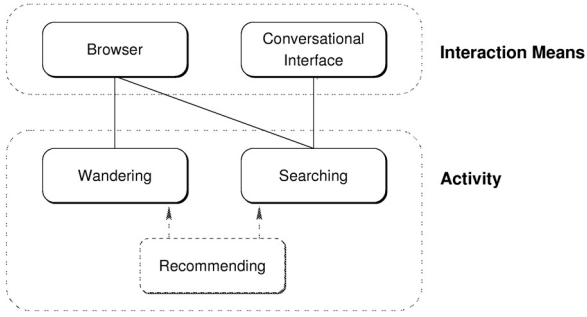

Fig. 1. User activities with photographs, and the proposed interaction means to provide computer assistance.

browsing assistant, and most importantly, the points where the two paradigms interconnect. Not only do they share the same user profile and the same domain database (the photographs and their meta-data), but the conversational search engine also signals the browsing assistant anytime the search space changes.

2.3 Use-Case Scenario

The following use-case scenario elucidates how the described system would be used in a real-world situation.

It's a lazy Sunday afternoon; Claire has picked up her portable photo browser, and is now sitting in the couch wandering through her photo collection. Over the years she has accumulated over ten thousand photographs—way more than what she would be able to handle without the aid of intelligent software assistance.

Claire ponders over a photo of the Sagrada Familia that she took in Barcelona many years ago. The browsing assistant allows her to see all related photographs across multiple dimensions: by choosing the contents *attribute, the system displays the thumbnails of all the other photographs which also contain the Sagrada Familia, closely followed by photographs which contain other cathedrals; choosing the* location *attribute presents her with all photos which were also taken in Barcelona, and likewise for the other attributes.*

At this point Claire remembers that she took a brilliant photograph of her pet dog Bello playing on a beach somewhere in Spain. She asks the system for some help in finding it: 'James, could you help me find a photograph? It was a photo of Bello on a beach'. The system—known as James by Claire—promptly reduces the search space to those photos containing both Bello and a beach. Claire can see this change immediately, because the browsing assistant only displays the photos which meet the specified criteria.

The search is not yet complete, however, as there are still roughly fifty photographs left. Thankfully, the system knows that Claire is usually quite good at remembering also the location of her photos. 'Claire, do you remember where it was taken?'. To which Claire provides a positive reply. 'The photo was taken in Spain'. At this point, the search space is reduced to a couple of dozen photographs, and the system determines that the

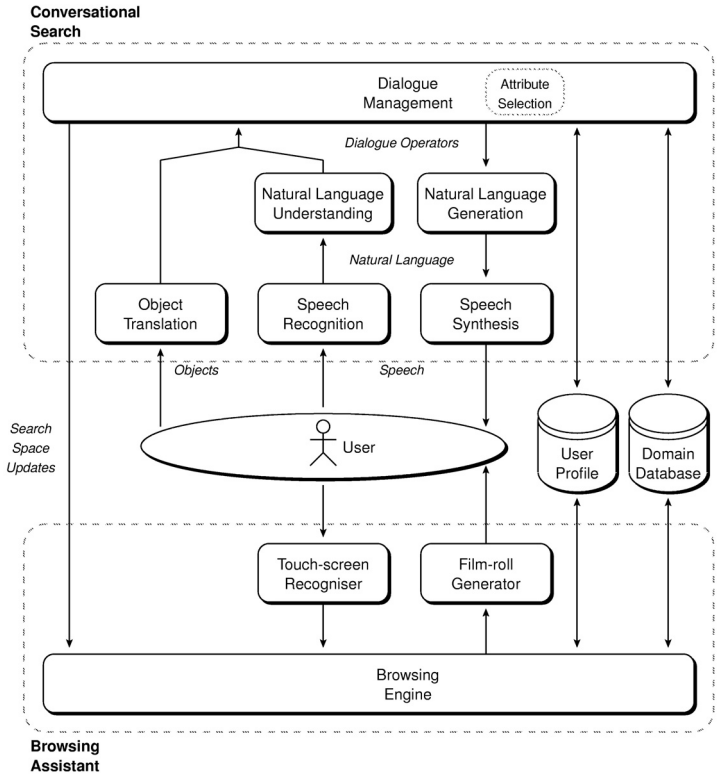

Fig. 2. The architecture of the system. One can see the major components of both the conversational search system and the browsing assistant. Also noteworthy are the contact points between the paradigms, and the central role played by the user.

date *attribute provides the best chances of reducing it further. 'Claire, do you remember when it was taken?', asks the system. Claire does not remember, but she is already satisfied with the current set of photos: 'Thanks James, and please stop!'*

3 Characterisation of the Photo Domain

Before we proceed to the interaction paradigms and the associated algorithms, it is important to understand the structure of the data we will be handling. In this section we will describe the details surrounding the meta-data about the photos.

There are several problems concerning the definition of a structure for the meta-data of domestic photographs. The most pressing issue is user-related: each person has a different perception of what matters about a photograph and how it should be classified. As an example, let us consider what attributes are needed to describe what is visible on a photograph. Most users would want an attribute like *people*, since that is one of the

focus points of anyone's life. However, if we allow *people*, why not also consider *pets*, or *monuments*, or whatever is important for different users? Even though this problem seems insurmountable without providing each user with their own personal meta-data categorisation, a middle-ground solution can be attained. We have classified the meta-data about each photo into five generic attributes; the internal structure of each attribute can then be tailored to suit the user's wishes.

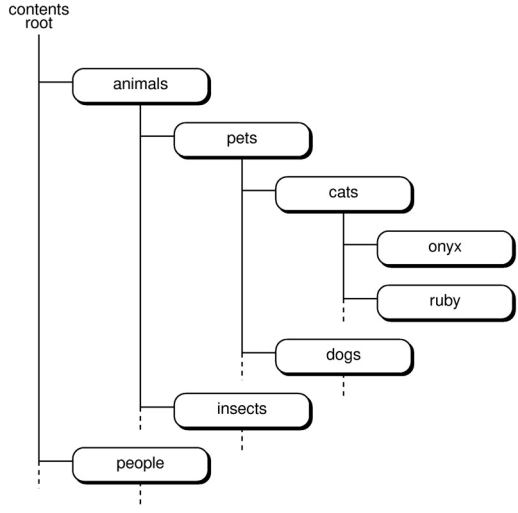

Fig. 3. A subset of the ontology tree for the contents attribute.

Table 1. The attributes constituting each photo's meta-data.

Attribute	Cardinality	Orderability	Structure	Similarity	Example
Contents	multiple	unorderable	hierarchical	partial match	/plants/flowers/tulips
Location	single	unorderable	hierarchical	partial match	/Europe/Spain/Barcelona
Topics	multiple	unorderable	hierarchical	partial match	/holidays/camping trips
Date	single	orderable	flat	temporal distance	21^{st} of June, 1999
Album	single	unorderable	flat	total match	"Summer 1999 in Spain"

The first attribute we considered, *contents*, stores all the information which is visible in the photograph, including categories such as *people* or *pets* which we have discussed previously. In order not to lose the conceptual division between the different categories, we have opted for the construction of an ontology tree capable of grasping the inter-connections between the various elements found in the real world (a subset of that tree

is shown in Fig. 3). Likewise, the *location* attribute also requires an ontology tree, in this case to store the physical location where the photograph was shot. The tree we used follows a division of the planet into continents, countries, regions, and cities. More refined branches are possible, of course, since in many cases users will be aware of those subdivisions. The rationale is to provide always enough structure that allows for any possible query from the user, e.g., 'show me photos taken in Africa', 'show me photos taken in Spain', 'I want to see photos from Montparnasse', etc.

The *topics* attribute represents the high-level category where the photo belongs, such as 'holidays' or 'parties'. It also requires an ontology tree because users might consider subcategories such as 'birthday parties' or 'camping holidays'. Simpler is the meta-data stored by the *date* attribute: it just contains the time stamp of when the photo was taken. Finally, the *album* attribute contains the single event where the photo belongs. In the days of 35mm film, it would correspond to one film roll or set of film rolls taken sequentially.

Table 1 provides more detailed information about the properties of the five attributes. The *cardinality* property indicates whether the attribute will have single or multiple instances for a photograph; *orderability* is the property that allows us to define comparison operators between instances; the *structure* will be hierarchical for those attributes which allow the definition of an ontology tree, or flat for those that do not; and at last the *similarity* column merely indicates which distance measure is used to compute the similarity between photos.

As we have explained in Sect. 2.1, photographs are a form of domestic content. This fact has one important consequence as far as the meta-data is concerned: contrary to what happens for publicly available content, there will be no professionally produced meta-data. Either the users have to manually annotate all their photographs, or we must rely on automatic techniques to produce it. The former possibility is not only unrealistic (especially considering the huge increase in the amount of available photos), but would also represent a contradiction as far as any vision of ambient intelligence is concerned. Fortunately, the latter possibility is becoming more and more feasible. For some attributes, such as the date and the location, modern digital cameras with built-in clocks and GPS receivers can already provide the meta-data. Regarding the contents, there is abundant research on automatic feature extraction techniques, showing promising results [9],[10].

4 The Browsing Assistant

The browsing assistant was implemented according to the general principles outlined in Sect. 2.2. In consonance with the requirement that the system should fit in a living room environment, the software was designed to run on a portable webpad, making use of the touch-sensitive screen for input. As one can see from the screenshot in Fig. 4, the main elements of the user interface are as follows: in the centre, a large area where the user can display a photograph for viewing; to the left, a panel with all the meta-data attributes, allowing the user to choose the preferred dimension for wandering (see Fig. 5); to the right, the film-roll with thumbnails of photographs related to the current one according to the selected dimension; at last, in the bottom, a text area displaying the meta-data of the current selection.

Whenever a new photograph is selected by being dragged from the film-roll onto the centre, the film-roll is updated with the thumbnails of the most similar photos according the currently selected dimension. Since only a limited number of thumbnails can be displayed, the user can scroll the film-roll in order to view the thumbnails of the second-best suggestions, the third-best, etc. Changing the current dimension is done by clicking on any of dimension icons. The thumbnail suggestions will immediately be changed in tune with the switch of dimension.

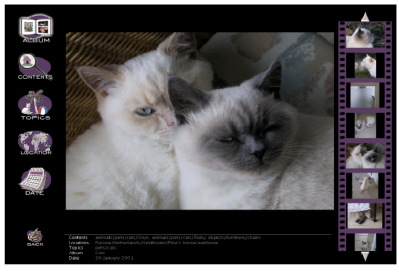

Fig. 4. The user interface of the browsing assistant.

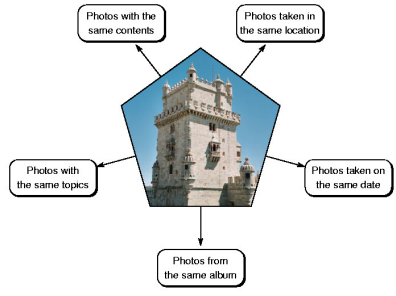

Fig. 5. The browsing assistant provides multiple dimensions for wandering. These correspond to the available meta-data attributes.

4.1 Computing Photo Similarity

In order to present the user with suggestions of photos somehow related to the one currently being displayed (which shall forth be named the *pivot*), one must determine which photos are good candidates. Moreover, since the number of thumbnails which can be displayed simultaneously must be small—not only because of limited screen-estate but also to avoid overwhelming the user—it is necessary to rank the thumbnails by order of similarity and to display the top matches first.

The algorithms used to compute the similarity measure differ according to the attribute (these are listed in Table 1). In the case of the album attribute, the comparison is straightforward: a photograph either belongs to the same album as the pivot or it does not. Also for the date attribute the calculation is simple: the similarity is simply the temporal distance between the two photographs.

The three hierarchical attributes present a bigger challenge however. The reason is that even non-perfect matches are relevant if they share part of the tree structure. Let us consider for example the contents attribute, and take a look at the photos stylised in Fig. 6. If photo A is the pivot, which of the other photographs should be ranked higher? Photo B contains two strong partial matches, but photo C contains one perfect match. In other words, should the algorithm take a generalist or a specialist bias? We believe that the user should make the ultimate choice. With this in mind we have constructed an

algorithm which can handle the partial matches of hierarchical attributes and allows for the fine-tuning of the generalist/specialist bias.

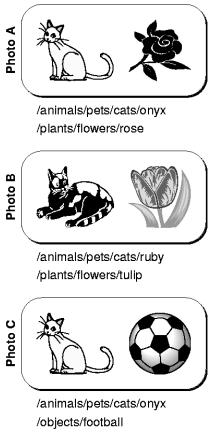

Fig. 6. The photographs used as examples for content similarity.

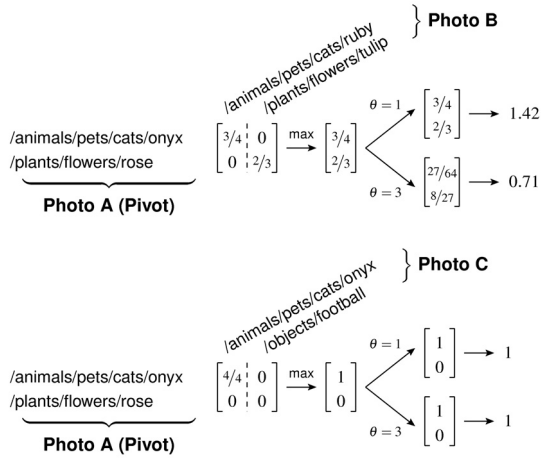

Fig. 7. The partial matches algorithm.

The *partial matches* algorithm works as follows. The first step is to construct a matrix where the row and the column headers are the meta-data of respectively the pivot and the photo to be compared. Each element in the matrix is computed as the fraction of the number of components from the test photo which match the pivot, divided by the length of the pivot tree. The next step is to build a vector where each element is the maximum value (the best match) of each of the matrix rows.

The following step is the key to the fine-tuning between the generalist and specialist preferences. Each element in the vector is raised to the power θ. This operation introduces a non-linear bias to the calculation, being the fractions least affected those close to the unity (which is not affected at all). In practical terms, the higher the value of θ, the higher the penalty imposed on partial matches, and the more specialist the bias. The last step of the algorithm consists of simply adding the elements of the vector: the obtained sum is the similarity measure.

Figure 7 shows the various steps of the algorithm, using photo A as the pivot and photos B and C as the test photos. Notice that in accordance to the above description, for $\theta = 1$, photo B is the most similar, while for $\theta = 3$, the more specialist photo C takes that role. Also noteworthy is the fact that the similarity measure is not necessarily commutative.

5 Conversational Search

We have defined conversational search as the process where a conversational interface is used to assist a search activity. As stated, the principle consists of having a dialogue between the system and the user as the means to construct a query iteratively and thus to progressively reduce the search space. Since people are so well-versed in conversation, one can deduce that it is critical for the system to engage in a dialogue which feels natural and consistent. Also, since the system plays the role of an all-knowing assistant, leading the conversation in the direction of a quick discovery of the user's target, it is likewise important that the system asks the right questions to the user. In this section we will discuss these and other issues essential for the good performance of a conversational search engine.

5.1 Dialogue Operators

Human language provides virtually infinite ways for people to express their intentions. As far as the conversational engine is concerned, all this diversity is unmanageable and largely redundant. For this reason, the engine does not process raw human dialogue (either in speech or text form), but rather the high-level dialogue operators associated with it. These are largely derived from research conducted on *speech acts* [11], and intend to capture the essence of each dialogue utterance. As depicted in Fig. 2, a number of low-level components—those handling speech recognition and synthesis, plus natural language understanding and generation—take care of the back and forth translation between speech and dialogue operators.

Table 2 lists some of the most important dialogue operators used by the system. These are classified into different categories according to the role they play in the dialogue. The core operators are the ones dealing with the *constraining*, *relaxation*, and *suggestion* operations. The meta-operators deal primarily with the control of the dialogue itself, and the clarification operators are mainly used to provide extra information to the user [7].

Table 2. A selection of the dialogue operators used by the system (*attr* represents the attribute, and *conf* the confidence level on the user response).

Operator	Category	Source	Parameters	Example
ASK_CONSTRAINT	constrain	system	attr	'Do you remember the location?'
ACCEPT_CONSTRAINT	constrain	user	attr, value, conf	'The photo was taken in Spain.'
REJECT_CONSTRAINT	constrain	user	attr, conf	'I don't remember.'
ASK_RELAXATION	relax	system	attr, value	'Perhaps it was not taken in Spain?'
ACCEPT_RELAXATION	relax	user	attr, value	'Yes, you are right.'
QUERY_VALUES	clarify	user	attr	'To which places have I been?'
START_DIALOGUE	meta	user	—	'Help me find a photo, please.'

5.2 Use of Objects

Using high-level dialogue operators to encode the user's intentions has another added benefit: the system becomes modality-agnostic, relying on any input/output translators we wish to implement. While speech was the first choice for the reasons we have already stated, it is possible to experiment with other modalities. In particular, previous experiences with object interaction within our project suggested that using physical objects as filters for photographs could also provide a natural and fun way for users to interact with the system [12],[13]. Moreover, research on graspable interfaces presents them as valid but still largely unexplored input modalities [14]. For all these reasons, we have also constructed an input translator for object interaction, as depicted in Fig. 2.

The hardware necessary for the translation is based on RFID (Radio Frequency Identification) technology, and was already available and ready-to-use. The basic idea consists of embedding small RFID tags in the objects we wish to identify, and to hide the detection coil beneath a table. Each tag has a unique identifier, allowing us to know precisely which objects the user places on top of the table.

The translator converts the low-level events of placing and removing objects from the table into the dialogue operators ACCEPT_CONSTRAINT and ACCEPT_RELAXATION, respectively. Each object also has an associated attribute/value pair from the photo database, enabling the user to express the equivalent of 'Show me photos taken in Barcelona' by simply placing her souvenir from Barcelona on the table. Since one can associate anything with an object, people and pets present themselves as obvious candidates. Furthermore, since the system recognises multiple objects, complex queries and filters can be constructed in a very intuitive and tangible manner: as an example, the user could add a puppet dog to the table to express her desire to see the photos of Barcelona that also contain her pet dog.

5.3 Dialogue Management

Dialogue management is the problem of handling the high-level dialogue intentions in such a way as to provide the users with the feeling that they are interacting with an intelligent entity. In practical terms, the dialogue manager must interpret the user dialogue intentions and select which dialogue operation to perform at each step.

In many telephony applications of conversational interfaces, the dialogue tends to follow a fairly rigid structure of question-answer sequences. Breaks to the sequence are allowed, but they are treated as exceptions that must be resolved before the sequence can proceed again. It is our view that this often called *ping-pong* dialogue structure does not fit well in the definition of an ambient intelligent conversational engine. Rather, one crucial aspect of dialogue management for in-home situations is what we call the *asynchronicity* of the dialogue. The system might still direct the communication and ask questions to the user, but these do not follow a rigid programme. The user is also free to ignore the system's questions, to provide answers out-of-sequence, or to provide information that was never asked. The rationale is that home environments tend to be fairly chaotic places, and the primary objective of the dialogue is to constrain the search space, not the user. Furthermore, the concept of using objects as an interaction modality would be defeated if the user were forced to use them only as a reply to a question from the system.

Concerning the implementation, the asynchronicity of the dialogue is achieved by giving the user enough time to provide multiple dialogue intentions in a single iteration, and by always reassessing the state before a new question is posed. Moreover, the dialogue manager does not rely on the user to immediately provide answers to questions asked. The heuristics that decide which operation to perform at any moment are straightforward and can be hard-coded into the system: while the search space is large, the decision is always made to *constrain* it; once we have narrowed it down to just a handful of elements, then we can *suggest* them to the user; should it happen that the search space becomes over-constrained—meaning that there are no photos which match all search criteria—then a decision is made to *relax* one of the constraints. It should be noted that even though plenty of research exists on the optimisation of dialogue strategies, most results suggest that the heuristics we use are optimal in these circumstances [15],[16].

5.4 Attribute Selection

Even after the dialogue management heuristics have made a decision regarding which operation is the most appropriate at any given moment, the system might have another problem to solve before it can ask a new question to the user. In particular, the operations which suggest to the user a constraint or a relaxation of the search space demand one extra parameter: the target attribute for constraining, or the constraint subject to relaxation, respectively. The problem of attribute selection lies precisely in the determination of this extra parameter. The decision must take a number of different factors into account, namely the characteristics of the search space, and the knowledge and preferences of the user.

Regarding the first factor, the analysis of the search space is based on the *maximum entropy method*, which in the case of a constraining operation chooses the attribute which maximises the potential reduction of the search space, or in the case of a relaxation operation, the constraint which increases it the least. This procedure is based on the standard formula from information theory for the calculation of entropy: $H(a) = \sum_{i=1}^{n(a)} -P_i(a) \times \log P_i(a)$, where $H(a)$ denotes the entropy for attribute a, and $P_i(a)$ is the probability of finding a value i belonging to attribute a, if we were to uniformly select a photo at random.

As far as the second factor is concerned, modelling the user is justified by the realisation that we can avoid asking the wrong questions (the ones which the user knows nothing about) if we have some insight about the typical behaviour of the user. Of the several facets that can be modelled, the ones most relevant for a conversational search system include the user's interests and the user's knowledge about the domain. The former assumes that people are most likely to search for items that they prefer[1]; in the case of the latter, the assumption is that people's knowledge is not uniform across all the domain's attributes and items. At the current time we have limited the profile to only keep track of the user's knowledge.

[1] User preferences could be accomodated by considering non-uniform item probabilities in the calculation of the entropy, for example.

5.5 The Objective Function

Particularly in the context of an ambient intelligence scenario, improving the quality of a dialogue system means foremost providing a better experience to the users. This realisation has a direct impact on the definition of the objective function which will assess the performance of the algorithms used for attribute selection.

Even though a thorough evaluation of user satisfaction can only be performed by extensive user testing, we believe that it is possible to determine beforehand a number of objective metrics which are likely to play a significant role in the overall user satisfaction. The most obvious is the total number of steps necessary to attain the final goal, the assumption being that users will prefer to locate their photographs as quickly as possible. Another possible metric considers that long dialogues might not be detrimental to user satisfaction as long as they feel that progress is being made. To be more explicit, this metric would consider *rejections*—defined as questions which the user is forced to decline because she does not have an answer for them—as being quite frustrating, and should thus be avoided, even if at the expense of a slightly longer (but safer) dialogue. Obviously, many other metrics are possible and even likely to be relevant. One could say that, for example, a single rejection is not harmful if intermixed with multiple positive responses. Again, only extensive user tests will provide a categorical answer to this question.

It should be noted that any entropy-based heuristic will attempt to minimise the length of the dialogue, while a profile-based heuristic strives to reduce the number of rejections. While there is a correlation between these two measures, it is not linear and far from obvious. A more detailed discussion on the issue can be found in [17].

5.6 User Profiles and Stereotypes

We have already commented on the potential benefits one can derive from knowing the typical behaviour of the user. However, one must also take into account that any profile-based heuristic will suffer from the well-known *blank slate* problem whenever a new user is brought into the system. Basically, since the heuristic will need time to adapt to the particularities of the user, the initial performance will be far from optimal. Using stereotypes provides an elegant solution to this problem, and was therefore one of the primary considerations when designing a framework for attribute selection.

The basic principle consists of having multiple profile heuristics in the system. One of them will be the user's personal profile, and all the others will represent a variety of different stereotypes. By designing the system in such a way that the importance of a heuristic automatically adapts in accordance to its performance, we can rely mainly on the stereotypes in the beginning, and later switch to the personal profile once it has learned enough about the user. Obviously, the expectation is that at least one of the stereotypes will be close enough to the user to provide good results.

Figure 8 depicts a framework for the process of attribute selection which includes support for stereotypes. Furthermore, it also enables the fine-tuning between profile-based and entropy-based heuristics, thus allowing one to adjust the objective function to suit the user. At last, it also enables the enforcement of policies which might run against the dictates of both the profile or entropy-based heuristics. As an example, consider an

application where a certain attribute a must always be asked first. Even though our conversational search engine for photographs does not make use of such policies, one could imagine other applications that require them.

Adaptation takes place by changing the weights of the profile heuristics. After each round, the heuristics which provided good suggestions are rewarded (their weights increase), while the others are penalised (their weights decrease). Eventually, the weight of a heuristic will reflect its probability of being right, i.e., its reliability. It is this adaptation mechanism which allows the smooth transition between the initial estimates based on stereotypes and the later ones relying on the user's personal profile. Again, a thorough description of this framework can be found in [17].

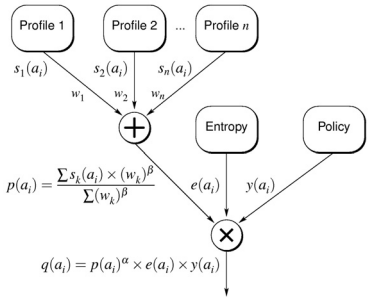

Fig. 8. The framework for attribute selection. Typically the system will have one personal profile heuristic and various profile heuristics based on stereotypes. The combination of the estimates of all profile heuristics is given by $p(a_i)$, the estimate of the entropy-based heuristic is defined by $e(a_i)$, and the suggestion from the policy is $y(a_i)$. The final 'quality' value of attribute a_i is given by $q(a_i)$.

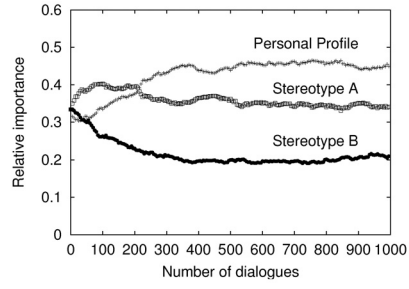

Fig. 9. Adaptation of the profiles. The advantages of using stereotypes can be clearly seen in this graph: they provide good results in the short term, compensating for the relatively slow learning process required by the user's personal profile.

6 Experimental Results

The evaluation of a system like the one described poses a number of challenges. While the browsing assistant can be assessed by conventional methods for user testing, the evaluation of the conversational search system is complicated by the reliance on user models: the importance of learning cannot be forgotten, and requires that the test be conducted over a long period. Furthermore, using stereotypes poses another issue: how can we construct enough of them to have a 'pool of stereotypes' large enough to be effective?

Nevertheless, fine-tuning many of the heuristics and parameters used in the framework can be done by relying on simulated users. A simulated user is a mathematical model which attempts to capture the essence of how a real person would behave. It has

the advantage that one is not limited by constraints such as user fatigue, boredom, and the relatively slow speed of real-time interaction. On the other hand, simulated users can be dangerous, and pose a number of potential issues that one must be aware of. Foremost, no matter how intricate the model, it will always fail to capture all the subtleties of a real person. The consequences will range from skewing the results towards the model, to the possibility of overfitting the data.

At the moment, the simulated users we implemented capture a single aspect of the user knowledge, namely the fact that each attribute will have a different probability of being answered positively by the user. However limited, it has enabled us to conduct preliminary tests on the feasibility of the proposed engine for conversational search. One of the most interesting results corresponds to the evaluation of the mechanism used to incorporate stereotypes: the outcome can be seen in Fig. 9. It represents the evolution of the relative importance of three different profile heuristics: a personal profile, a relatively close stereotype A, and a more distant stereotype B. As one can see, the advantage of using stereotypes is obvious: in the short term they are indispensable, quickly providing good results, and compensating for the fact that the personal profile needs some time to learn about the user. The adaptation mechanism also comes out in evidence: as the number of dialogues increases and the personal profile improves its performance, the weights slowly adapt to reflect this change. As a result, the personal profile takes the predominant role in the long term. Also, as one could expect, stereotype A is better than stereotype B.

Obviously, one cannot rely on simulated users forever, and as such, one of the next steps in our research will be to evaluate the presented system in a real-world setting.

7 Conclusion

In this paper we have taken a look at photo viewing in home environments. We have argued that given the present increase in the number of photographs that people take, the issue of content overload will soon be brought into the domestic arena. With the goal of mitigating this problem, we have first identified the sort of activities that people engage in with their photographs. We have then proposed computer assistance in the form of two interaction paradigms: conversational search and the browsing assistant. Furthermore, we have shown how these paradigms could be brought under the umbrella of an unifying architecture, providing the user with a seamless experience.

The core of the paper focused on a detailed description of each of the two interaction paradigms and how they integrate with one another. One of the most important conclusions we derived was that user modelling plays a prevalent role in a system that aims to be ambient intelligent. On that note, part of our future work will address precisely the improvement of the user models used by the conversational search engine. Moreover, given the importance of the user side of the equation, work of this nature will never be validated without thorough tests with real users. This also shall be addressed soon.

Acknowledgements. The authors would like to thank the members of the PHENOM project team: Esko Dijk, Elise van den Hoven, Nick de Jong, Evert van Loenen, and Yuechen Qian.

References

1. Aarts, E., Marzano, S., eds.: The New Everyday. 010 Publishers, Rotterdam, The Netherlands (2003)
2. Maes, P.: Agents that reduce work and information overload. Communications of the ACM 37 (1994) 31–40
3. Resnick, P., Varian, H.R.: Recommender systems. Communications of the ACM 40 (1997)
4. Rodden, K., Wood, K.R.: How do people manage their digital photographs? In: Proceedings of the ACM Conference on Human Factors in Computing Systems (ACM CHI 2003), Fort Lauderdale, Florida, USA (2003)
5. Lieberman, H.: Autonomous interface agents. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (ACM CHI 1997), Atlanta, Georgia, USA (1997)
6. Zue, V.: Conversational interfaces: Advances and challenges. Proceedings of the IEEE 2000 (2000)
7. Langley, P., Thompson, C., Elio, R., Haddadi, A.: An adaptive conversational interface for destination advice. In: Proceedings of the Third International Workshop on Cooperative Information Agents, Uppsala, Sweden (1999)
8. Göker, M.H., Thompson, C.A.: Personalized conversational case-based recommendation. In: Proceedings of the 5th European Workshop on Case Based Reasoning, Trento, Italy (2000)
9. Rui, Y., Huang, T.S., Chang, S.F.: Image retrieval: Current techniques, promising directions and open issues. Journal of Visual Communication and Image Representation 10 (1999) 39–62
10. Wang, J.Z., Li, J.: Learning-based linguistic indexing of pictures with 2-D MHMMs. In: Proceedings of the 10th ACM International Conference on Multimedia, Juan-les-Pins, France (2002) 436–445
11. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge Unversity Press (1969)
12. van Loenen, E.: On the role of graspable objects in the ambient intelligence paradigm. In: Proceedings of the Smart Objects Conference, Grenoble, France (2003) 3–7
13. van den Hoven, E., Eggen, B.: Digital photo browsing with souvenirs. In: Proceedings of Interact2003, Zurich, Switzerland (2003)
14. Fitzmaurice, G.W.: Graspable User Interfaces. PhD thesis, Dept. Of Computer Science, University of Toronto (1996)
15. Levin, E., Pieraccini, R., Eckert, W.: Using markov decision process for learning dialogue strategies. In: Proceedings of ICASSP98, Seattle, Washington, USA (1998)
16. Litman, D.J., Kearns, M.S., Singh, S., Walker, M.A.: Automatic optimization of dialogue management. In: Proceedings of the 18th International Conference on Computational Linguistics (COLING-2000), Saarbrucken, Germany (2000)
17. Teixeira, D., Verhaegh, W.: Optimising attribute selection in conversational search. In: Proceedings of the Sixth International Conference on Text, Speech, and Dialogue—TSD 2003, České Budějovice, Czech Republic (2003)