Dramasearch. Character-mediated search in Cultural Heritage.

Emanuele Borini, Rossana Damiano, Vincenzo Lombardo and Antonio Pizzo

Dipartimento di Informatica and CIRMA, Università di Torino

Abstract— The Dramasearch project imports the Dramatour methodology for character-based guided tours into interactive search applications for web sites and data repositories. The virtual character acts as a mediator between the user and the search engine, without replacing but complementing the role of the search engine. The character interacts with the user through a multimodal interface, with the aim of assisting the user to formulate her/his query in an effective way, presenting her/him the results and recommending related resources.

In this paper, we describe an architecture for character-mediated search applications. In particular, we describe the process of dramatizing the interaction with the user, a feature that is required to qualify the character as a cooperative agent, and we show how the domain expertise of the character, embedded in the architecture as a semantic search layer, is exploited to construct the coherence of the interaction.

I. INTRODUCTION

A wide range of recent multimedia applications, from virtual assistants to video-games, exhibit sophisticated approaches to information presentation that involve multimodal aspects and character-based mediation. Artificial characters improve the naturalness of the interaction with the user, making the system appear more responsive and cooperative [13], [2]. In cultural heritage, embodied conversational characters have been employed as virtual guides for cities [3] and exhibitions [14]. Multimodality characterizes as well most of the applications developed for the field of cultural heritage [4], [18], [19].

The working assumption of the Dramatour methodology is that a character, who acts in first person and shares the visitor's present time and space, yields a powerful effect of physical and emotional presence [5]. The emotional properties of drama improve the effectiveness of content reception [15] and the user engagement, bringing an affective element to her experience [17].

In this paper, we describe an architecture for character-mediated search applications inspired by the Dramatour methodology. In particular, we describe the process of dramatizing the interaction with the user, a feature that is required to qualify the character as a cooperative agent, and we show how the domain expertise of the character, embedded in the architecture as a semantic search layer, is exploited to construct the coherence of the interaction.

II. CHARACTER-BASED PRESENTATIONS IN CULTURAL HERITAGE

The Dramatour approach merges production methods for dramatic media [8] and formal annotation techniques to build information presentation systems. The assumption underlying the methodology is that the dramatization of the exposition enhances the effectiveness of the communication through the users engagement in the emotions displayed by the characters [17]. The emotions of a character result from the conflicts she/he engages with her/himself or with other external entities during the drama performance. In a plot designed by a drama author, such conflicts increase in number and intensity until they find some resolution [16]; so, the drama features a rising then falling emotional course (often called dramatic arc [10]).

The application for character-based guided tours 'Carletto the virtual guide' was designed by following the Dramatour methodology [5]. In April 2006 the application was opened to the public for one week in the historical location of Palazzo Chiablese in Turin. The presentation, delivered on a mobile device, was generated by selecting and sequencing the clips in a way that accounted for the position of the user in the historical site. The semantic coherence of the presentation was guaranteed by the fact that the system follows the relations encoded in the ontology to generate the presentation. The evaluation performed on the visitors revealed a positive acceptance of the character and of the application. In 'Carletto the virtual guide', the presentation was segmented into atomic audiovisual units, annotated according to the topic they deal with, and assembled on-the-fly in response to the users input according to a presentation script compliant with the drama tenets sketched above. The characters presentation addressed both the task of providing information to the visitor and the task of building a bond with the visitor for emotional engagement.

The Dramasearch project imports the Dramtour methodology in search application. It relies on three main assumptions: the believability of the character, its capability to support the user in her/his search for information, its ability to retrieve and deliver the resources that match the user query.

- The believability of the character is obtained by incorporating dramatic qualities into the interaction with the user, so as gain her/his emotional engagement of the user as the interaction proceeds. Beside the task of supporting the user in the search task, the character must be designed so as to show a personality, for example, by joking or giving information about her/himself from time to time.
- The character is able to interpret and focalize the user queries by using an ontological representation of the domain. In the example application we refer to in this paper, the domain is given by a baroque palace, that is conceptualized along a set of different dimensions, according to the hierarchical structure that characterizes

light-weight ontologies. The ontology is useful not only to interpret the user query in a semantically aware representation of the interaction context, but also to help the user focus or enlarge her/his query and to propose related topics to continue the interaction.

- The character relies on a search engine to satisfy the user queries, using the ontology to drive the search operated by the search engine. According the metaphor of the virtual assistant, the character uses the search engine as a tool, that is explicitly mentioned in the interaction, to which the character acts as a mediator who exploits her/his domain knowledge to post the right queries to the engine.

The combination of the elements mentioned above (dramatized interaction, domain ontology, search engine) is aimed at obtaining an application that overcomes the drawbacks of the individual components, obtaining a virtual character who interacts with the user without displaying the limitations that characterize the commercial implementations of assistive characters; a search application that delivers the results according to a semantics shared with the user, and is able to assist the user in the search, allowing her/him to 'navigate' the domain along the ontology.

III. ARCHITECTURE

When the interaction begins, the character greets the user and introduces her/himself by qualifying her/himself as a virtual assistant and providing motivations for her/his help. The reason is that, in order to gain believability, the character must display a socially aware behavior, so the social aspects of the interaction have priority over the informative ones.

When the user inputs a query, the system extracts the keywords from it, based on a list of predefined keywords. The keywords provide the link with the ontology: every node is associated with a set of keywords. In case no keywords can be extracted from the user query, the system asks the user to input a new query. The system calls the search engine on the user keywords and classifies the resulting resources with respect to the ontology that represents its domain knowledge. The subsequent behavior depends on the size of the result set. If the size is below a certain threshold, the character presents the results by describing them according to the ontology; when needed, he/she asks for further information; then, the character suggests new topics to the user. If the size is above the threshold, the character suggests to the user possible ways to modify the query. When the user abandons the interaction, the systems acknowledges it appropriately.

The character displays an evolution along the interaction with the user, becoming friendlier as the interaction proceeds. The degree of friendliness affects the character's proactivity and the form of her/his communicative acts: at the beginning, the character's attitude is very formal, becoming more elliptical and informal in the subsequent interaction.

The architecture of the system (Figure 1) is conceptually divided into three layers:

- The **recommendation layer** is responsible for helping the user to formulate an effective query, i.e., a query that outputs a 'small' set of results, with a threshold configured during the design of the application. Or, if the search outputs no results, this layer suggests the user a new topic. It contains the *Recommendation Manager* and the *Search Engine*; its functioning deeply relies on the knowledge encoded in the *Domain Ontology*.

  In order to account for alternative conceptualizations of the domain, the domain knowledge can be constituted by a set of ontologies. By doing so, the same resource can be classified based on its relations to different ontologies, much in the same way as resources can be accessed based on different facets in web sites. In the system knowledge base, each resource is described as a tuple of ontology nodes [20].

- The **dialogue layer** is responsible for analyzing the user input in order to identify the communicative act performed by the user and to select the appropriate continuation (the next communicative act).

  The communicative competence of the systems relies on two knowledge sources, i.e., a library of *Communicative Acts* (inspired by FIPA [9]) and a list of ontology-related *Keywords*. It contains the *Dialogue Manager* and the *Comparison Module* for managing the keyword-ontology match.

- The **interface layer** is responsible for the realization of the communicative acts of the character. In fact, what is considered to be a communicative acts for the system at an abstract level can be mapped onto different realizations, conveyed to the user through various modalities. For example, a proposal can be communicated verbally according to different formulations or non-verbally by displaying a set of option buttons.

  The surface realization of the character's communicative acts is handled by the *Interface Manager*.

The functioning of all three layers depends on the current context of the interaction, stored in the *Interaction History*. The Interaction History is a record of the past interaction, i.e., the list of the communicative acts performed by the system and the user, including the user queries and the system recommendations, if any. By consulting the Interaction History, the system modules are able to exclude from consideration the topics that have already been tackled and the resources that have already been proposed to the user. Moreover, the Interaction History allows the system to compute the parameters that affect the character's behavior, like self-confidence and friendliness.

A. Interface Manager

The Interface Manager receives a Communicative Act type and a content item (one or more topics or an informative unit) from the Dialogue Manager and selects the communicative modality given the current context of the interaction; the latter is encoded in the Interaction History. For examples, the Dialogue Manager may contain rules that associate the earliest phases of the interaction with the verbal modality, and an already well–grounded interaction with the non-verbal ones.
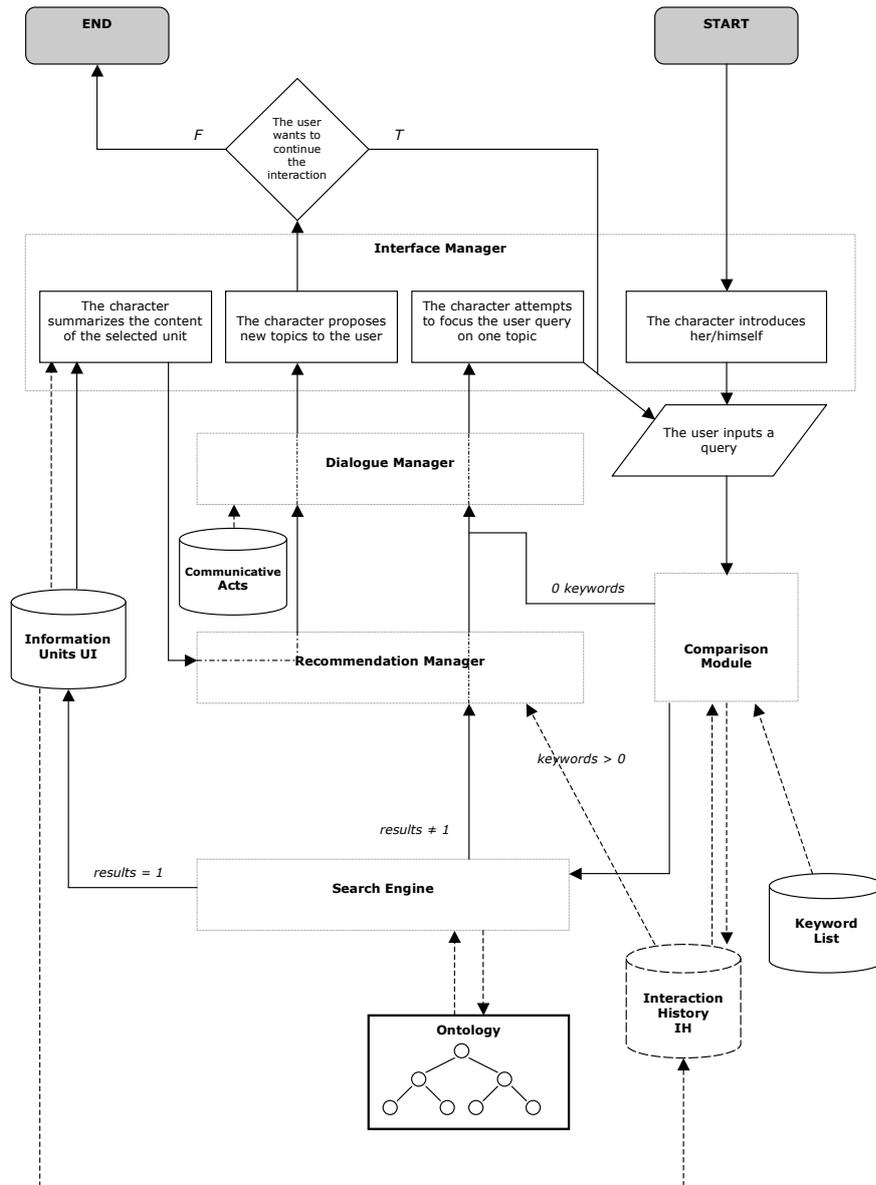
Fig. 1. The system architecture of the Dramasearch project. Solid lines represent control flow, dashed lines represent data flow.

Once a modality has been selected, the Interface Manager maps the Communicative Act type onto an actual realization that is appropriate given the character's current attitude. For example, if the character's previous suggestions have been accepted before, the character's higher degree of confidence may lead to a more assertive attitude, i.e. one in which proposal are expressed directly ("I suggest you ..." instead of "May I suggest you ... ").

This module is implemented as two separate sets of rules for handling the selection of the modality given the communicative act type and the selection of its realization given each modality. Verbal and graphical templates provide the realization of the communicative act types.

The Interface Manager also takes care of communicating to the user the type of the resources found (video, text, image, etc.), by using the information attached to the resources themselves.

B. Dialogue Manager

When the user inputs a query, the Comparison Module extracts the keywords from the query based on the Keyword List, if any. The Dialogue Manager identifies the Communicative Act type of the user query by combining its linguistic

form and the keywords it contains with the expectations set by the discourse context, encoded in the Interaction History.

At the beginning of the interaction, the Dialogue Manager selects the appropriate Communicative Acts to let the virtual character greet the user and introduce her/himself. Conversely, when the interaction ends, it selects the Communicative Acts for thanking and good-byeing the user as well. From time to time, depending on the character design, it may insert some socially–oriented Communicative Acts based on the features of the interaction encoded in the Interaction History (duration, friendliness). Although it constitutes a tenet of the character-base methodology, we do not detail this aspect here, since it is tightly related to the design of the specific application.

The keywords mentioned by the user are relevant for the analysis since, in some cases, they allow disambiguating between different Communicative Act types. For example, the user may be requesting to perform a new search (a *request*) or may be accepting a suggestion (an em accept-proposal) issued by the character: when the keywords extracted from the user query match the topic suggested by the system, the user input counts as an acceptance act. From the viewpoint of dramatization, the two situations mentioned above need different reactions by the character: the character may become more confident as a consequence of the acceptance of her/his recommendations. Note that the Dialogue Manager is not directly concerned with the execution of the Communicative Acts, but simply sets appropriate variables that are subsequently tested by the Interface Manager.

After analyzing the user input, the Dialogue Manager inputs the keywords to the Search Engine. The Search Engine performs the search with the *and* operator. The search results are filtered by the Recommendation Module and sent back to the Dialogue Manager, which deals with the problem of selecting the appropriate Communicative Act given the results of the user query. The results of the search can consist in a resource, in a request for disambiguation, or in a recommendation. If no resources are found, the system can suggest a new query, depending on the degree of proactivity assigned to the character by its designer.

The Dialogue Manager reacts to the results of the search according to a set of rules that map the input from the Recommendation Manager onto Communicative Act types

- If there are no results, the character notifies the failure of the search (*failure*) and asks (*request*) the user to input a new query.
- If there is 1 item to present to the user, the character tells the user that an item of a certain type has been found (*inform*).
- If the Dialogue Manager receives from the Recommendation Manager a request to disambiguate among different referents for the user query (in case of multiple resources found), the characters asks the user to indicate a preference for one (*query–ref*).
- If the Dialogue Manager receives from the Recommendation Manager a request to recommend a topic, the character suggests to the user a topic that may be of

interest for her/him (*propose*).

C. Recommendation Manager

The Search Engine outputs a set of items, ordered according to their relevance and classified according to their relation with the ontology. In order to do so, the ontology nodes are tagged with the relevant keywords, i.e., each node is associated with the set of synonyms that can be used to refer to that concept. One synonym is tagged as the preferred one for use when the system needs to refer to that node as a possible topic in generating a recommendation.

As anticipated before, resources are classified according to different ontologies, in order to grasp the fact that the same domain can be conceptualized in different ways. So, an item is normally related to different nodes, posited along different ontologies, and its content is formally described as a tuple of nodes, corresponding to the different ontologies. Ontologies are prioritized to reflect their importance in conceptualization.

The ontological knowledge is exploited by the Recommendation Manager for two main purposes. First, the relations between nodes and ontologies allows it to distinguish items which share the reference to the same node or nodes, since it is improbable that they share the same classification with respect to all the ontologies. For example, if items A and B refer to the same node x in ontology 1 but to different nodes y and z in ontology 2, the ontology 2 allows the system to distinguish them and to describe them effectively. Second, the Recommendation Manager uses the ontological knowledge to make recommendations to the user, by suggesting the user to restrict or to enlarge the scope of the search, or recommending resources that share some topics with the retrieved ones (a technique adopted also by [1]).

The Recommendation Managers removes from the results the items that, according to the Interaction History, have already been proposed to the user. Then, it reacts differently depending on the size of the result set:

- No items found. Note that this case is possible only if the keywords in the user query are at least two: if the user inputs only one keyword, that keyword is directly matched against the keyword list; if the lookup succeeds, it means that there is at least one related resource linked with the corresponding ontology node. With multiple keywords, it is possible that the search engine does not retrieve any resources by performing a search with all the keywords. In this case, the system calls the search engine replacing the *and* operator with the *or* operator and examines again the obtained result set.
- 1 item found. The Recommendation Manager passes the item to the Dialogue Manager for presentation.
- N items, $N < T$, where T is a threshold set by the application designer. In this case, we assume that the artificial character, differently from the standard search engine, should try to guide the user to identifying one relevant topic at a time, as it happens in conversation, where topics are normally addressed one by one. So, the

character considers the user query as ambiguous and asks the user to indicate a preference for one of the items.

The Recommendation Manager passes to the Dialogue Manager the description of the items (a set of tuples referring to the domain ontologies, with a tuple describing each item), so that the Dialogue Manager can describe them effectively when asking her/him to indicate a preference for one of them. In fact, if the items share the reference to one or more nodes, the Dialogue Manager can rely on the nodes they do not share for constructing a definite description that effectively distinguish them. The items are ordered according to their specificity (i.e., the depth of the nodes they refer to, where the depth is considered as the depth of the lowest–level node in the tuple describing the item) and the predefined ordering among the ontologies.

- If the item number is above the threshold T, the Recommendation Manager exploits the ontology to restrict the scope of the user query, in the following way:

  1) The Recommendation Manager classifies the items according to the priority ordering over the ontologies, so that the items that are related with the highest-priority ontology are considered before than those that have no relation with it.
  2) For each item, it examines the tuple of nodes that describe it and considers the node situated in the highest priority ontology (*candidate node*). With the aim of identifying a set of more specific 'topics' (here, nodes) that are susceptible to generate a smaller result set, the Recommendation Manager selects the child nodes of the *candidate node*.
  3) If the number of the child nodes is below a certain threshold (established by the application designer), these nodes are passed to the Dialogue Manager, so that it can propose them as new topics for the next user query. Otherwise, the next ontology in the priority order is examined, and so on. Notice that the refinement is non recursively extended to the child nodes of the candidate nodes, so as to keep adherence with the query input by the user.
  4) If no appropriate topics can be found, the Recommendation Module issues a failure, that will be dealt with by the Dialogue Manager.

- Finally, the system is able to give the user recommendations following a successful search.
  In order to do so, the Recommendation Manager tries to extend the search to the items that share some topic (here, node) with the items resulted from the last executed search. This operation corresponds to a 'focus shift' in discourse terminology [12], normally attempted when the current topic has been explored to a sufficient degree.

  1) In order to find related topics, the Recommendation Manager considers the tuple describing the item that has been just proposed to the user. For each node in the tuple, it selects its siblings in the ontology taxonomy, if any, and adds them to a list of *candidate topics*. If number of siblings is below a certain threshold M, they are discarded as they are likely to generate a large set of results.
  After examining the entire tuple, if the list of candidate topics is empty, it the Recommendation Manager considers the most direct ancestor of the each node in the tuple and adds it to the list.
  2) The nodes that appear in the Interaction History (i.e., they have been already mentioned as topics) are discarded and the list of the *candidate topics* is ordered according to the preference ordering over the domain ontologies.
  3) Finally, it calls the Search Engine on the list of the *candidate topics*. Querying the search engine is necessary in order to make sure that the suggested topics are susceptible of giving any results at all. However, only the topics that yield less than T items (where T is the threshold set for user-driven queries), if any, are recommended to the user.

IV. PROTOTYPE APPLICATION

In this section, we provide an example of the proposed architecture. The example is taken from an application that is currently been developed as a prototype. This application re–uses the repository of informative units that constitute the knowledge base of 'Carletto the virtual guide', the character–based system for guided tours on mobile devices mentioned in Section II and described in [5].

The Information Units which constitute the knowledge base of 'Carletto the virtual guide' are encoded as self-contained audiovisual clips in which an artificial character, the anthropomorphized spider Carletto, presents specific information about a historical site. The domain of the presentation consists of a historical location situated in Turin, Palazzo Chiablese, a 16th Century baroque palace that hosts the former royal apartments of the Savoy family. The royal apartments include five rooms: the Guardroom, the Room of the Valets, the Reception Room, the Dining Room, and the Room of the Tapestries. A domain expert (an historian of the art) has provided the information to be conveyed to the users. The input from the domain expert has been encoded in a set of micro-scripts by a drama expert, from which the audiovisual clips containing the presentation delivered by the artificial character have been produced, with the help of a multimedia production team. The description of the domain has been encoded in an ontology with the assistance of an ontology engineer, providing the basis for the semantic description of the units. The ontololgy is described in more detail in [7].

The prototype we use here for exemplifying the functioning of the architecture shares with of 'Carletto the virtual guide' both the repository of Information Units and the ontology created for the mobile application. Since the Information Units often refer to the character of Carletto, the character is also reused by the prototype.

In the following, we briefly describe the ontological description of the domain, originally designed for the mobile application, and provide an example of the abstract architecture described in Section III.

A. Ontology

Since the location of Palazzo Chiablese is a complex domain, that can be described according to several semantic dimensions (like history, art or topology), the representation of the domain is subdivided into five specialized ontologies (see Figure 2), and the topic of the presentation units is described as a tuple of references to these ontologies. This representation accounts for the fact that the same unit possibly concerns more than one topics, each described by a different ontology. A unit is constrained to refer to only one class or instance in each ontology, but may not refer at all to a certain ontology (provided that at least one ontology is referenced).

- The *topological ontology* describes the topology of location, centered on notions like rooms and room parts, according to the practice normally followed by human guides. In the mobile application, it is the reference ontology.
- The *historical ontology* describes the historical facts related to the location. This ontology includes two main branches, describing respectively the historical characters who lived in the palace and those who worked in it (further subdivided into painters, architects and craftsmen).
- The *ontology of objects* systematizes the variety of pieces of furniture and other items located in the apartments, most of which are awkwardly termed and unknown to standard visitors. It is the reference ontology in web applications.
- The *chronological ontology* is an ontology of time intervals that serves the purpose of providing a temporal framework for locating the historical events.
- The *symbolic ontology* describes the concepts (reigns, battles, marriages) that are celebrated by the art objects, located in the palace (paintings, statues, ect.).

In these ontologies, concepts are connected by subsumption relations in which each concept has only one ancestor, so all the ontologies are taxonomies. In order to simplify the description, the same concept cannot appear in more than one taxonomy.

In order to represent the non-taxonomic relations among concepts, orthogonal relations (different than subsumption) have been added to the domain description to connect concepts within the same ontology or across different ontologies. For example, the topological ontology contains a taxonomy of topological concepts (like rooms and room types) and the *part-of* relations according to which these entities are related, specifying, for instance, that a room contains a set of walls, a ceiling and a floor. As an example of relations spanning across different ontologies, consider the *subject of* a painting (object ontology), *located in* a room (topological ontology) and *painted by* an artist (historical ontology) to *celebrate*

an event (symbolic ontology). The domain ontologies have been developed with the Protégé ontology editor; orthogonal relations have been implemented as Protégé slots [11].

In the search application, the repository does not include the units devoted to the management of the interaction with the user (for example, the units in which the virtual character greets the visitor – without providing any domain information).

B. Example

1) When the interaction begins, the virtual character Carletto greets the user and introduces himself, establishing his role of museum guide (the Dialogue Manager passes the related Communicative Acts to the Interface Manager, and the Interface Manager maps them to the contextually appropriate verbal and graphic realization). For example:

    "Carletto: Hello, my name is Carletto, and I will take care of you during your visit. Do you want to begin your tour in Palazzo Chiablese?"

2) The user greets Carletto and asks him a question, keying her/his request in a field. Example:

    User: "Hello, my name is xxx, and I want to know (...)"

3) The Interface Module calls the Comparison Module to examine the words keyed by the user; they are matched with the Keyword List in search for keywords. The Communicative Act of the user is recognized by the Dialogue Manager as a *request* to perform a search with the given keywords.

    - If no keywords are matched, the system activates the Dialogue Manager to generate some communicative act aimed at inducing the user to ask something about Palazzo Chiablese in general. The Interface Manager communicates this to the user. Example

        User: "Hello, I want to know who cooked the good cake I had in the coffee shop in the street".

        Carletto: "Dear xxx, I'm very happy that you have this familiarity with me and you want to chat, but actually I'm here to be your guide... There are a lot of interesting objects in this palace. It's ok for you if we talk about Palazzo Chiablese?"

    - If there are one or more keywords matched, the Dialogue Manager forwards them to the Search Engine.

4) The search can yield 0, 1 or more results.

    - The Search Engine returns 0 results when there are keywords into the words keyed by the user, but there aren't any resources whose content can be described by all the user keywords at the same time (more precisely, there are no resources that are related to the entire set of the ontology nodes associated to the user keywords), since the search is operated, by default, with the 'and' operator. In response to the 0 results, the Dialogue Manager decides to communicate a *failure* to the user,
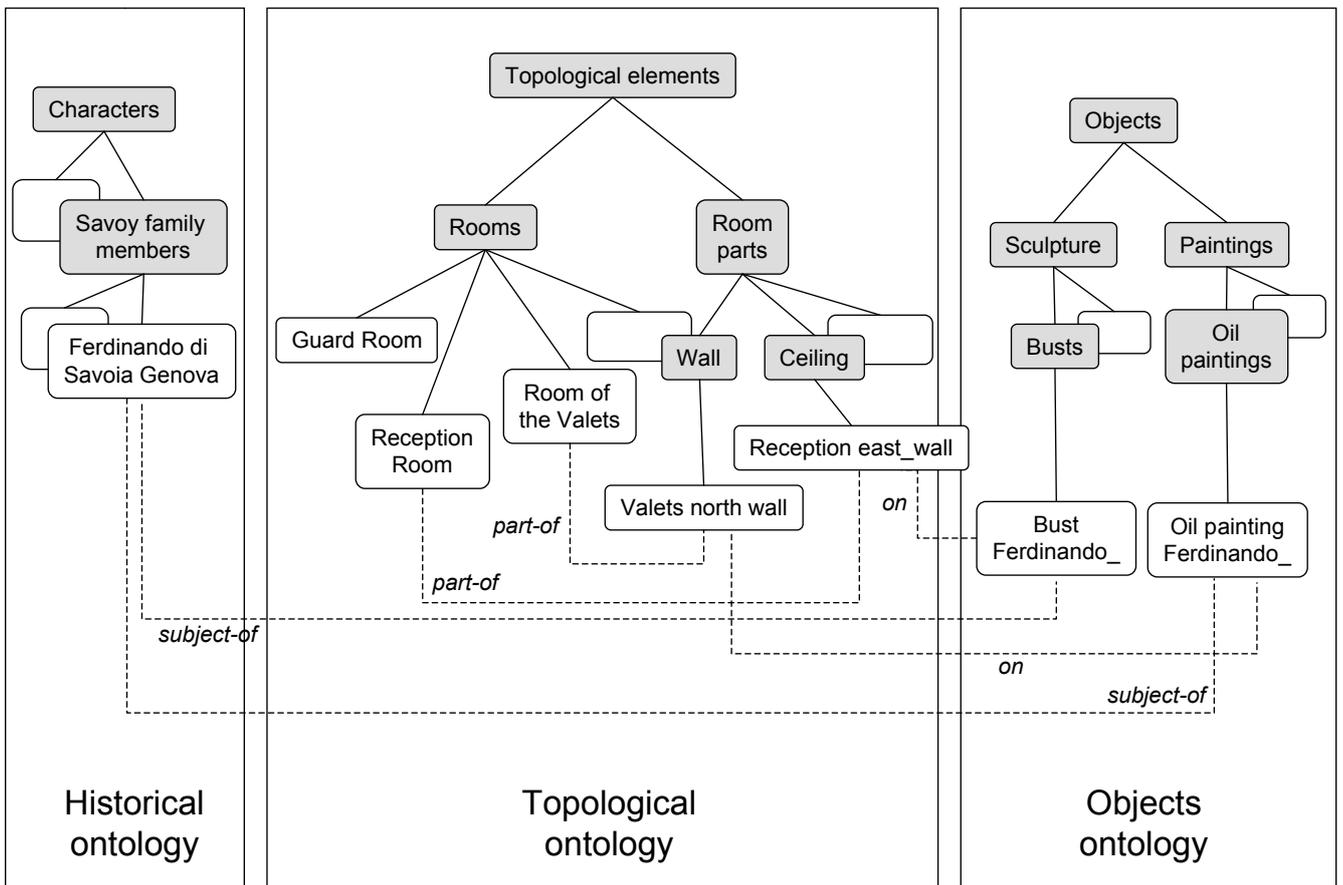
Fig. 2. A fragment of the domain ontology concerning Palazzo Chiablese, including portions of three specific ontologies (historical, topological and object ontologies). Grey boxes represent classes, white boxes represent instances. Subclass and instance relations are represented by solid lines, non-taxonomic relations are represented by dashed lines.

letting the Interface manages select how to communicate this to the user. Example:

User: "Talk me about Beaumont and the fireplaces"

The Comparison Module matches two keywords: 'Beaumont' and 'fireplace', but the Search Engine doesn't find any resources with these keywords (i.e. no resources is related with both nodes corresponding, respectively, to 'Beaumont' and 'fireplace'). However, the Recommandation Module prompts the Search Engine to perform the search with the 'or' operator, obtaining some results.

Carletto: "I think you are a little confused... There are no relation between Beaumont and a fireplace! If you want, I can talk about many pictures by Beaumont or many fireplaces, but not about both at the same time."

- The Search Engine returns 1 result if only exactly one ontology node is linked to all the keywords matched by the Dialogue Manager (in case more than one Information Units are related with this node, the Recommendation Manager select the one with highest ranking, where the relevance rank is given by the Search Engine). The Recommendation Manager passes the Information Unit

found to the Dialogue Manager, which selects an *inform* Communicative Act and passes the task of generating it to the Interface Manager. Example:

User: "Talk me about the above–doors by Beaumont"

The Comparison Module matches two keywords: 'Beaumont' and 'above–doors', and the Search Engine finds only one resource linked to these 2 keywords.

Carletto: "In the 4th room there are some above–doors painted by Beaumont. They represent the Virtues".

- The Search Engine returns many results if there are many ontology nodes linked to the keywords matched by the Comparison Module.
  The system activates the algorithm for $results < T$ and passes the results to the Dialogue Manager, which selects a *query–ref* Communicative Act to let the user indicate her/his preference. The Interface maps the communicative act selected by the Dialogue Manager onto a realization (a verbal answer or a set of radio buttons). Example:

User: "Talk me about the works by Beaumont contained in the Palace"

The Comparison Module outputs 2 keywords: 'Beaumont' and 'Palace', and the Search Engine finds 4 items linked to these keywords (corresponding to four different instance nodes in the object ontology).

Carletto: "In this Palace there are many Beaumont works about which I can talk to you: the painting 'Alessandro Magno sulla tomba di Achille' in the 1st room, the paintings 'Battaglia di Canne' and 'Morte di un guerriero' in the 2nd room, the above-doors in the 4th room. Which of them would you like to know about?"

- The Search Engine returns too many results $results > T$ if the keywords matched by the MC are linked to high-level ontology nodes. The system activates the algorithm for $results > T$ and passes the results to the Dialogue Manager. The virtual assistant tries to induce the user to refine his request. Example:

User: "Talk me about this Palace"

The Comparison Module outputs 1 keyword: 'Palace', and the Search Engine finds 92 results. The system discover, through the algorithm, that the search can be breaked up in three different topics: 'interiors', 'pictures' and 'sculptures'.

Carletto: "Dear xxx, you make me a question too generic! Do you prefer I talk about the interiors, the pictures or the sculptures?"

5) The system suggests new topics to continue the conversation. Example

User: "Tell me something about the painting representing Pia de Tolomei".

The Comparison Module outputs 2 keywords: 'painting' and 'Pia de Tolomei' and the Search Engine gives 1 result. Given the *inform* Communicative Act selected by the Dialogue Manager, the Interface Manger exposes the content of the Information Unit found by the Search Engine.

Carletto: "You see... on the north wall of the 2nd room... It's a classic scene: the death of Pia de Tolomei. She's supported by the ladies and the monk. It's a work painted in 1854 by Felice Bauducco"

The algorithm for finding related topics selects a set of items which share the same topic as the proposed resource along the ontology representing the topology of the Palace. The recommendation is passed to the Dialogue Manager, which selects a *propose* Communicative Act:

Carletto: "On the north wall of the 2nd room the are two interesting paintings: the picture representing Villa della Regina and the picture representing Via Po. Do you want me to talk you about one of them?"

V. FUTURE WORK AND CONCLUSIONS

In this paper, we have presented an architecture for building character–based search applications, in which the character act as a mediator between a search engine and the user and exploits a semantic representation of the domain to assist the user in focalizing and expanding her/his queries. The use of the artificial character relies on the assumption that the emotional impact of dramatization, incorporated in the Dramatour methodology [6], improves the involvement of the user and the acceptance of the application.

The architecture described in this paper, currently at an early stage of implementation as a prototype, requires further design and testing to become a full-fledged application. In particular, the recommendation system must be evaluated with real users, and an strategy to deal with alternative modalities in the communication with the user must be defined.

REFERENCES

[1] L. Aroyo, R. Brussee, P. Gorgels, L. Rutledge, N. Stash, and Y. Wang. Personalized museum experience: The rijksmuseum use case. In *Museums and the Web*, San Francisco, CA, 11-14 April 2007.

[2] R. Aylett, M. Vala, P. Sequeira, and A. Paiva. Fearnot!–an emergent narrative approach to virtual dramas for anti-bullying education. *LNCS*, 4871:202, 2007.

[3] N. O. Bernsen and L. Dybkjaer. Domain-oriented conversation with h.c. andersen. In *Proceedings of ADS 2004*, pages 142–153, 2004.

[4] C. Callaway, E. Not, A. Novello, C. Rocchi, O. Stock, and M. Zancanaro. Automatic cinematography and multilingual nlg for generating video documentaries. *Artificial Intelligence*, 165:57–89, 2005.

[5] R. Damiano, C. Gena, V. Lombardo, F. Nunnari, and A. Pizzo. A stroll with carletto. adaptation in drama-based tours with virtual characters. *User Modeling and User-Adapted Interaction*, 2008.

[6] R. Damiano, V. Lombardo, F. Nunnari, and A. Pizzo. Dramatization meets information presentation. In *Proceedings of ECAI 2006*, Riva del Garda, Italy, 2006.

[7] R. Damiano, V. Lombardo, F. Nunnari, and A. Pizzo. Ontological domain coding for cultural heritage mediation. In L. Lesmo and S. Borgo, editors, *Proceedings of FOMI 08*, volume 174 of *Frontiers in Artificial Intelligence and Applications*, Amsterdam, 2008. IOS Press.

[8] M. Esslin. *The Field of Drama*. Methuen, London, 1988 (1987).

[9] FIPA. FIPA ACL message structure specification. Technical Report XC00061, Foundation for Intelligent Physical Agents, 2001.

[10] Gustav Freytag. *Technique of the drama, an exposition of dramatic composition and art*. S.C. Griggs and Company, Chicago, 1985.

[11] J.H. Gennari, M.A. Musen, R.W. Fergerson, W.E. Grosso, M. Crubézy, H. Eriksson, N.F. Noy, and S.W. Tu. The evolution of Protégé: an environment for knowledge-based systems development. *International Journal of Human-Computer Studies*, 58(1):89–123, 2003.

[12] B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12:175–204, 1986.

[13] L. Ieronutti and L. Chittaro. Employing virtual humans for education and training in X3D/VRML worlds. *Computers & Education*, 49(1):93–109, 2007.

[14] S. Kopp, L. Gesellensetter, N. Kraemer, and I. Wachsmuth. A conversational agent as museum guide - design and evaluation of a real-world application. In *5th International Working Conference on Intelligent Virtual Agents (IVA'05)*, 2005.

[15] B. Laurel. *Computer as Theater*. Addison Wesley Longman, Reading, MA, 1993.

[16] R. McKee. *Story*. Harper Collins, New York, 1997.

[17] R. Picard. *Affective Computing*. MIT Press, 1997.

[18] A. Solon, P. McKevitt, and K. Curran. Mobile multimodal presentation. In *ACM MM'04*, New York (NY), USA, 2004.

[19] O. Stock and M. Zancanaro (eds.). *Multimodal Intelligent Information Presentation*. Springer, 2005.

[20] J. van Ossenbruggen, A. Amin, L. Hardman, M. Hildebrand, M. van Assem, B. Omelayenko, G. Schreiber, A. Tordai, V. de Boer, B. Wielinga, J. Wielemaker, M. de Niet, J. Taekema, M. van Orsouw, and A. Teesing. Searching and annotating virtual heritage collections with semantic-web techniques. In J. Trant and D. Bearman, editors, *Museums on the Web*, San Francisco, CA, 11-14 April 2007.