Multimedia Tools and Applications (2019) 78:13613–13648
https://doi.org/10.1007/s11042-019-7362-5

FANTASIA: a framework for advanced natural tools and applications in social, interactive approaches

Antonio Origlia[1] · Francesco Cutugno[2] · Antonio Rodà[3] · Piero Cosi[4] · Claudio Zmarich[4]

Received: 10 May 2018 / Revised: 5 February 2019 / Accepted: 11 February 2019 /
Published online: 21 February 2019

Abstract

With the recent availability of industry-grade, high-performing engines for video games production, researchers in different fields have been exploiting the advanced technologies offered by these artefacts to improve the quality of the interactive experiences they design. While these engines provide excellent and easy-to-use tools to design interfaces and complex rule-based systems to control the experience, there are some aspects of Human-Computer Interaction (HCI) research they do not support in the same way because of their original mission and related design patterns pointing at a different primary target audience. In particular, the more research in HCI evolves towards natural, socially engaging approaches, the more there is the need to rapidly design and deploy software architectures to support these new paradigms. Topics such as knowledge representation, probabilistic reasoning and voice synthesis demand space as possible instruments within this new ideal design environment. In this work, we propose a framework, named FANTASIA, designed to integrate a set of chosen modules (a graph database, a dialogue manager, a game engine and a voice synthesis engine) and support rapid design and implementation of interactive applications for HCI studies. We will present a number of different case studies to exemplify how the proposed tools can be deployed to develop very different kinds of interactive applications and we will discuss ongoing and future work to further extend the framework we propose.

Keywords Human computer interaction · Game engines · Application development tools

✉ Antonio Origlia
antonio.origlia@unina.it

Extended author information available on the last page of the article.

1 Introduction

In the last few years, industry-grade game engines such as Unity 3D[1] and the Unreal Engine 4[2] (UE4) have been made available to the general public. Differently from previous experiences, which required strong programming skills to actually make use of the engines, the newest versions of these tools pay extreme attention to entry-level users, developing user-friendly interfaces for developers to access their most powerful technologies. As a matter of fact, the interest of game engine developers is now to make features such as artificial intelligence (AI), environment navigation, animation logic, etc... more accessible and easy to integrate into complex workflows. In this situation, such tools cannot be simply considered rendering engines, as their capabilities go much beyond simple image generation and are capable of handling complex mechanics and different input modalities. Moreover, the growing communities of developers, together with the industrial push towards new technologies, are making modern game engines an important alternative to consider when developing interactive applications for research purposes.

With the success of game-based entertainment, the industrial push towards innovative interactive experiences has brought significant technological advances. Moreover, the increasing amount of small game development studios, caused by the availability of electronic stores reaching a worldwide audience at small costs, has also generated an interest towards providing high-end game engines with development interfaces that greatly speed up the development process. While Unity 3D has opened the way to this approach, in the last years, UE4 has gained significant attention as, with respect to UE3, a significant rework of the editor's interface has been proposed. This testifies to an important interest, on the side of the industries, towards providing ease of access to complex technologies to the community of developers. This situation also has implications for researchers working in the HCI field.

Unity opened the way to the exploration of the possibilities offered by freely available game engines in the HCI field. Its applications now range from augmented reality experiences for chemistry learning [11] to the creation of virtual physics laboratories [9]. Following Unity's success, the fourth instalment of the Unreal Engine editor was also designed to be distributed to a wider audience and is becoming frequently used for HCI research, too. Recent works on cultural heritage, for example, used it to create virtual museums [13] and, in general, there is an interest in integrating such technologies into interaction studies, for example in the fields of autonomous driving simulation [30] and computer vision [28].

First of all, the fast-paced advances in the design of interactive experiences on an industrial level must be matched by the academy in order to avoid newly developed interfaces being damaged by comparison with the extremely competitive design of digital entertainment applications. In the HCI field, where interface design is crucial, dealing with users' expectations may become a serious challenge as the gaming industry continues to grow as "Entertainment gaming experiences color players' interactions with other digital media, setting expectations for user experience and interactivity" [32]. Since the industry itself provides the technology needed to support this design process by making it easily accessible, connecting the development of applications designed for academic purposes to the advances of industrial solutions helps researchers not to fall behind when the perceived quality of

[1] https://unity3d.com/

[2] www.unrealengine.com

an interactive system depends also on comparison with the standards set by the gaming industry.

From another point of view, the constellation of innovative interaction devices, ranging from body motion sensors to eye trackers and haptic devices, is as fluid and rapidly changing as the technology behind gaming engines. Integrating these devices separately into a completely independent architecture often requires effort and, once again, the risk of falling behind with respect to the evolution of these devices becomes high, as their development speed is high. Industry-level game engines, however, are equipped with plugin systems and provide precise guidelines to manage the integration of third-party technologies. Device producers are interested in converging towards popular game engines to maximise their reach towards the community of developers. By following the engines' integration guidelines, they also align to a consistent way to provide access to interactive functions. For these reasons, connecting the development of academic applications directly to game engines reduces the integration efforts needed to access new interactive devices and helps keep the development process consistent.

While the industrial effort concentrates on interface quality, development speed and technological integration, researchers in the HCI field are mainly interested in the models underlying the interactive processes. These aspects are less explored by the industry, making the main activity of HCI researchers complementary to the industry activity. Integration between the two efforts has, in our opinion, the potential to bring benefits to both realities. Specifically, the practical advantages of designing an approach centred on freely available technology the continuous evolution of which is efficiently managed by the industry lie in the possibility, for researchers, of concentrating on interaction management models, rather than on technicalities brought in either by the necessity to compete with the design quality standards set by the entertainment industry or by the need to integrate a new control device. Of course, we do not claim that our approach is necessarily adequate to support any possible interactive experience that may be designed. We concentrate, instead, on supporting the development of interactive experiences that are based on natural interaction and have an impact on social processes. In this view, a framework of reusable code to integrate game engines with tools that are important for studies on natural interaction is important, to support HCI research. The foundations of this effort can be summarised by the following points of interest:

- Build upon the interest of the gaming industries in providing advanced environments to develop interactive experiences.
- Exploit the convergence of device manufacturers towards game development platforms to avoid the need to deal with integration technicalities.
- Let researchers concentrate on interaction management models while avoiding being left behind with respect to industrial standards set on design.

In this paper, we illustrate the process of design and development of a toolset, named FANTASIA, to expand the capabilities of a game engine with an extensible set of modules, each specialised on a specific domain, designed to support HCI research on various levels and to allow rapid development of interactive applications. While each specific module provides specialised support introducing well-defined features, FANTASIA is designed to make the integration of these features fast and coherent with the normal design process of the included modules. Thus, FANTASIA lets developers take advantage of cutting-edge, well-maintained technology to offer extendable editors that make access to additional features as easy as access to native features. For each module we selected to be part of FANTASIA, we produced specific tools that allow integration and simplify development.

Each of these tools follows the design directives of the specific module to let developers access the features introduced by FANTASIA in a seamless way with respect to the normal development process offered by the module. FANTASIA is, therefore, linked to the modules we chose to implement the toolset. While this may appear to reduce the possibilities of scientific exchange, the modules we selected are designed to be extended and adapted to very different situations: from this point of view, we believe that much of the flexibility that is lost by considering specific modules while developing FANTASIA tools is compensated by the openness of the selected modules towards extension and modification. In the rest of this paper, we will refer to existing software specialised on a specific domain and selected for use in FANTASIA, as *module*. We will, instead, refer to the software add-ons we developed to integrate such modules and support fast development workflows as *tools*.

The paper is organised as follows: in Section 2, we summarise previous approaches our work can be compared with and describe the main differences characterising our solution with respect to the reference ones. In Section 3, we describe the tools FANTASIA provides to integrate the selected modules and different kinds of interactive experiences. In Sections 5.1, 5.2 and 6, we present the applications on which FANTASIA tools have already been used to create interactive experiences in order to demonstrate its versatility. This paper extends the work presented in [20], where we limited our discussion to the architecture presented in Section 5.1.

2 Related works

Research about new HCI approaches often concentrates on improving the efficiency of the transfer process between the abstract representation of proposed interaction modality and the actual implementation. For example, the TOUCAN framework [17] supports the development of Java applications by keeping code connected to the intended interaction models. While this approach mainly focuses on supporting the validation step of the developed applications with respect to the considered task models, it also implies the possibility of reusing code and speeds up applications development, which, however, is a problem more strictly linked to the development process itself. The complexity that lies in the development of interactive applications requires a significant effort, when designing them, to manage low-level technicalities that are more related to the technology than to the actual research interests. In particular, integration efforts are an important issue when a research group needs to investigate experiences sharing the same theoretical core but needs ad hoc solutions for the actual deployment. In this situation, an integration framework named Cognitive Environments Library for Input/Output [31] has been proposed to provide APIs that integrate different technologies through a managed communication channel. However, these approaches typically concentrate on communications layers, to which different actors, in an interactive system, must subscribe to exchange data. In this situation, developing low-level code is still necessary to implement the application. In our view, instead, the high-level development languages provided by game engines, but also by other specialised solutions, offer an important chance to simplify the process if they are directly integrated in a proposed framework, as we are doing with FANTASIA.

One research area on interactive applications which is of interest for our work is concerned with natural interaction, possibly involving social processes. Specialised frameworks thought to manage this kind of interaction and focus mainly on virtual human management are presently available. In these frameworks, game engines have been sometimes used but only as rendering modules. Modern game engines, however, are serious candidates to host most of the behavioural logic and the realisation modules in an integrated solution. The amount of possibilities offered by these advanced software tools poses a question with respect to the externalisation of services that the game engine would be able to provide itself. SAIBA (Situation, Agent, Intention, Behaviour, Animation) solutions like Greta [19] or Smartbody [34] do not rely on third-party modules and this indeed has its advantages. While including a significant amount of advanced research, however, these platforms often struggle in providing interfaces that are competitive with industrial standards and do not provide development frameworks that are as easy to use and up to date as the solutions now offered by the industry. This is normal, as it is not the main goal of the research teams behind these remarkable tools. However, the risk of them becoming outdated increases as research needs lead to concentration on other aspects of the platforms.

Mechanics and presentation are naturally intertwined in game development so that game engines offer very efficient ways to connect the internal status of the software with the front-end interface. This includes, of course, heads-up displays but it also includes virtual avatar behaviour. In our approach, the architecture of an industrial game engine is at the core of our approach and is integrated with platforms providing specialised tools to manage domains that are typically not important for game development such as, for example, dialogue management, which in games is reduced to conditional tree navigation. The same observation holds, in general, for probabilistic reasoning and knowledge representation, too. These platforms explicitly become part of our approach, which then includes high-level languages designed to simplify the development process. While this, indeed, leads to reduced flexibility with respect to approaches such as Smartbody and Greta, it has significant repercussions for development speed, tools dissemination and technological obsolescence.

The presence of a module specialised on dialogue management makes our framework close to the ones designed to support the development of chatbots. Both major actors of the industrial scene (e.g. Google DialogFlow, IBM Watson and Microsoft Bot Framework) and open source developers (BotPress and RASA) have been recently concentrating on simplifying the creation of such agents because of their clear impact typically on customer management. These frameworks, however, mainly concentrate on natural language understanding (NLU) and generation (NLG) and do not provide any specific integration with UI frameworks or knowledge representation approaches. Also, they typically assume interactions between one user and one agent. The architecture we present in this paper goes beyond dialogue management only and supports a more coherent design process across the different modules we selected, as functionalities offered for dialogue management and knowledge representations are already integrated among themselves and with the high-level programming approach offered by the game engine. No assumptions are made about the number of users and the number of agents involved in the interaction: interaction with small groups or with groups of agents can be implemented without altering the development process.

Summarising, differently from other approaches, FANTASIA relies on third-party applications to include specialised modules to develop solutions to specific parts of the development of natural user interfaces. The reduced flexibility is compensated by selection of modules equipped with plugin systems, so that extensions are supported. FANTASIA tools consist of code developed following the guidelines set by the plugin systems of the selected modules, so that modules integration and additional features are part of a consistent system and benefit from industrial development. The developed tools can be used to build integrated systems for natural interaction management by introducing features that are not usually provided by game engines. This way, a large, technologically advanced software architecture designed to meet high industry standards can be efficiently used by

HCI researchers to investigate more research-oriented aspects of interaction with reduced low-level programming effort.

3 The FANTASIA toolset

In this section, we briefly describe the *modules* we selected for the development of FANTASIA and explain our choice to include them. We then describe the FANTASIA *tools* that have been created to extend the functions of the selected modules and allow easy integration among them. We will also discuss how FANTASIA tools transparently make the specialised features of each module available without breaking the development approach characterising each module and how it connects the different modules to each other. All chosen modules provide mechanisms to allow easy extension of their features. FANTASIA combines the selected modules to provide a high-level interface to allow fast integration and provide general design guidelines. These emerge from the specific capabilities of each module and help developers to assign the management of different aspects of interactive systems to the modules that are optimised to handle them. This generates a wide variety of possible applications that can take advantage of up-to-date technology, rely on capabilities that are not available to each module, taken by itself, and be developed in an easy way. The role of FANTASIA in integrating the chosen modules is shown in Fig. 1. The proposed tools manage low-level communication between the considered modules and allow the use of high-level languages (XML for Opendial, Blueprints for UE4) to be used to manage interaction. This way, the typical development workflow of both Opendial and UE4 is used consistently, speeding up the development process while providing new functions. An overview of the FANTASIA tools is summarised in Table 1.

From a conceptual point of view, developing an application with FANTASIA requires operating on three levels. On the knowledge representation level, the available domain knowledge is organised independently from the task. On the interaction management level, the domain knowledge, the system's and the user's states are evaluated to compute the system's moves on an abstract level. On the front-end level, abstract instructions are evaluated with respect to the simulated environment and organised in a coherent presentation. In a typical application, UE4 would collect user input and update the Opendial dialogue state. Opendial would then estimate the correct system reaction, possibly accessing the domain knowledge represented in Neo4j, and instruct UE4 on how to proceed. For actions requiring

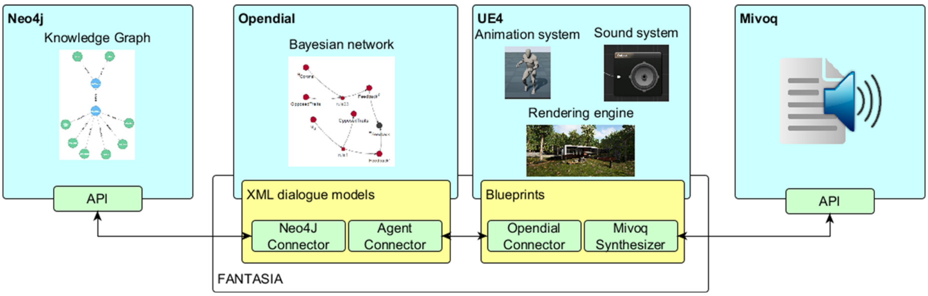

Fig. 1 The role of FANTASIA in integrating the chosen modules. Provided tools allow communication and service requests by exposing interfaces that integrate in the normal development process provided by each module

Table 1 Summary of the FANTASIA tools

Opendial	Neo4jConnector	Provides an interface for Opendial to access the knowledge base represented in Neo4j
	AgentConnector	Manages the connection between Opendial and the remote actors, mainly found in UE4
UE4	OpendialConnector	Manages incoming Opendial messages Sends back dialogue state updates as key-value pairs
	MivoqSynthesizer	Manages the connection with the synthesis engine Manages the memory dedicated to synthesised utterances Synchronises audio playback with lipsync data

a spoken answer, the Mivoq engine can be used by UE4 to obtain both audio and lipsync data.

The separation between abstract planning and physical realisation of behaviours is consistent with the SAIBA approach. However, in FANTASIA, knowledge representation and abstract planning are explicitly separated, while in the SAIBA specification they are integrated in a specific module named *intent planner*. With this approach, FANTASIA separates associative and deductive processes, which can be simulated through the data retrieval models provided by an NOSQL database, from predictive and decision processes, which are, instead, implemented by a dialogue manager. Another difference with the SAIBA framework lies in the unified representation of behaviour planning and realisation into the game engine. This choice is motivated by the strong connection that must exist between the *behaviour planning* module and the *behaviour realisation* module in order to provide real-time feedback to the user and to rapidly adjust to changes in the virtual world. Interface control approaches found in game engines, such as animation systems, have direct access to the simulated environments and to the external sensors monitoring the users' behaviour, thus allowing a fast, optimised way to continuously adjust the system's interface as needed. This is particularly important in the case of virtual characters, where coherent social behaviour heavily relies on prompt reactions to dynamically evolving situations, as in the case, for example, of mimicking strategies.

In the following subsections, we describe in detail the chosen modules used to build FANTASIA and what motivated our choices. We then describe the tools developed to integrate these modules and discuss how researchers can use FANTASIA to concentrate on developing interaction management processes and reduce the programming effort caused by technical needs.

3.1 Knowledge representation

The first module we selected for integration in FANTASIA is a graph database engine dedicated to domain knowledge representation. Specifically, we selected the Neo4j database. Neo4j [39] is an open source graph database manager that has been developed over the last 16 years and applied to a high number of tasks related to data representation [7], exploration [8] and visualisation [12]. It can be deployed in server mode and queried over a specific port using standard HTTP or the dedicated Bolt protocol. It can also be embedded in Java applications through dedicated APIs. In Neo4j, nodes and relationships may be assigned *labels* that describe the type of object they are associated with. Neo4j is characterised by high scalability, ease of use and its proprietary query language: Cypher. Cypher is designed to be a *declarative* language that highlights patterns' structure using an SQL-inspired *ASCII-art syntax*. A brief overview of the syntactic elements of Cypher queries is given here to help understanding of the example queries presented in this paper. The reader is referred to the online Cypher manual[3] for a more detailed presentation. In graphical representations of graphs, nodes are usually represented by circles, but in Cypher, nodes are represented by round brackets. For example, the query "MATCH (n:PERSON) RETURN n" returns all the nodes of the graph labelled *persons*. In the same way, since relationships are usually represented by labelled arrows in graph schemas, relationships between nodes are described by using ASCII *arrows*, too. The query "MATCH (m:PERSON)-[:KNOWS]-¿(:PERSON name: 'Andrea') RETURN m" returns all the nodes that contain people who know someone named Andrea. The SQL-like WHERE clause may be used to filter results using boolean logic. It is also possible to use Cypher to detect paths of specified length between nodes by using the * operator in the relationship definition.

In FANTASIA applications, encyclopaedic knowledge about the domain of interest should be represented in a Neo4j graph. The readable format of Cypher allows easy generation of queries for modules outside the database manager. Also, the overall system can take advantage of the optimisation offered for graph-traversing queries. Being a service module, Neo4j already provides interfaces to allow external connections through HTTP or Bolt protocols. Therefore, no extension for this module is needed: FANTASIA includes a solution to integrate Neo4j features client-side only for Opendial to submit queries and retrieve results. The client accessing the knowledge base hosted in Neo4j is a dialogue manager.

3.2 Dialogue manager

As already highlighted, game engines do not offer advanced solutions to develop probabilistic evaluation of the current state, being designed to mostly handle rule-based interaction. Therefore, they are limited when it comes to dealing with, for example, uncertain input or open conversations. This, however, is a frequent situation in HCI research, where natural interaction in noisy conditions is of interest. In designing FANTASIA, we decided to keep abstract reasoning separated from the game engine to take advantage of tools specifically developed to handle this kind of situation: dialogue managers. In the specific case of FANTASIA, we selected the Opendial [16] framework, which is based on probabilistic rules that merge the best of rule-based and probabilistic dialogue management. In cases where a good amount of previous knowledge about the domain is possessed by the dialogue designer with specific needs of fine-tuning rules, the rule-based approach can be integrated with probability and utility-based reasoning to fine-tune the system's response. This is profoundly different from the approach adopted, for example, by most frameworks for chatbot design, where solutions are strongly based on machine learning only. Probabilistic rules in Opendial are used to set up and update a Bayesian network consisting of variables that represent the current dialogue state. Depending on this, dialogue management works by selecting the most probable user action given a set of, possibly inaccurate, inputs and then computing the most useful system reaction, possibly generating natural language responses or executing actions. The dialogue domain is described in XML, following the Opendial specification, and it consists of a collection of models triggered by variable updates and rules that compose the model and influence the dialogue state.

[3] https://neo4j.com/developer/cypher-query-language

Since we concentrated on interactive approaches involving natural and social user inputs, it was important, during the development of FANTASIA, to support situations where uncertainty management and predictive capabilities are needed to successfully handle multimodal, noisy input and estimate the best course of action to maximise an arbitrary objective function. The approach proposed by Opendial with the use of probabilistic rules adds flexibility to FANTASIA as known management strategies provided by domain experts can be directly described in the domain. At the same time, probabilistic reasoning can be used for user adaptation and action planning when sources of uncertainty are involved.

3.3 Front-end interface

The game engine of choice to provide user interfaces in FANTASIA is UE4. This is one of the most famous game engines available and it has been used for over 20 years to develop many AAA titles. Following the success of other engines such as Unity, its source code has been made available and its editors have been significantly improved, with respect to previous versions, to allow entry-level users to take advantage of its many advanced features. Among the advantages of using UE4, other than its graphical capabilities and its optimisation for multi-user experiences, we considered the large availability of UE4 plugins distributed by the developers of advanced input devices (eyetrackers, VR goggles, etc...) and 3D development software, the continuous technological improvements produced to satisfy the needs of a highly competitive industry, the availability of tools to manage and combine many different kinds of multimedia content and the presence of a powerful visual programming language, which is also open to third-party extensions. Concerning this last aspect, while it is possible to work at code level with UE4 using C++, a high-level interface has been provided to simplify the composition of game logic and actors' control. Specifically, UE4 introduces a visual programming language called Blueprint, designed to let game developers describe control logic by composing a large set of *nodes* into control graphs. This makes it very easy for the final users to implement interaction logic and it also lets plugin developers provide access to the features they introduce in an intuitive way. The UE4 tools we developed as part of FANTASIA are all accessible via Blueprints in the relevant control graphs.

Among the motivations that led us to choose UE4 as one of the FANTASIA modules, its animation system is worth mentioning. UE4 offers different ways to animate characters that can be combined to create complex control systems. Tools provided by the engine include state machines, mono- and bi-dimensional blendspaces and per-frame evaluation logic. The Blueprint programming language is used to create complex movement drivers by selecting and blending together animations with a high degree of customisation. In the case of research-oriented tasks, this is of extreme importance as the management of complex non-verbal behaviour is a critical aspect concerning the naturalness of interactive agents. Through graphs, hierarchical systems and blending logic can be efficiently built inside the engine, thus optimising the development and rendering process. In this view, the approach of using completely externalised agent representations such as Smartbody that need to be copied by the engine does not allow developers to take full advantage of powerful real-time animation tools that are also directly tied in the system managing the virtual world and monitoring the users.

3.4 FANTASIA tools for dialogue management: knowledge access and dialogue state updates

Opendial is designed to keep track of the dialogue state and simplify the definition and the probabilistic tuning of models controlling the interaction. However, offering tools to host and manage encyclopaedic knowledge about the domains of interest, which may be very large, is not in its scope. For this reason, FANTASIA provides a tool to integrate the Neo4j querying process in the dialogue domain design approach typical of Opendial. Specifically, by including the FANTASIA Neo4j plugin in their dialogue domains, developers may use a special variable to submit their queries. Coherently with the way Opendial manages dialogue state updates, the tool detects updates on the special variable and, as soon as a new query is written in the monitored variable, it manages the submission to the database and recovers the result, in JSON format, as soon as it is available. Once the query result is received, the tool puts it into a second special variable. Developers, at this point, may continue developing their dialogue domain by introducing models that are triggered once the special variable FANTASIA uses to present the query results is updated. Using FANTASIA, developers never have to change the way they would normally work with Opendial, as variable monitoring and updating is the basic mechanism proposed by the framework and it can simply concentrate on Cypher query generation and response analysis to manage interaction informed by domain knowledge. This allows consistency to be maintained in the development process of the task management part.

The role of the dialogue manager is to generate abstract instructions to be executed by artificial agents. To simplify connections with remote agents, FANTASIA provides an Opendial tool to manage the connection with these agents and allow them to update the dialogue state when necessary. Both synchronous and asynchronous updates are supported by Opendial. The FANTASIA tools implementing these features do not make assumptions about the nature of the remote agents, which may be robots or 3D avatars, for example, and listens on open ports for incoming messages. The execution of the dialogue manager is suspended until all agents required by the current dialogue model have registered. Once all expected agents connect to the FANTASIA tool by specifying their ID, the dialogue manager may send instructions by specifying the ID of the destination agent through a further special variable. To allow Opendial to receive and evaluate updates from the remote agents, the FANTASIA tool accepts key-value pairs, which are used to create or update the corresponding variables in the Bayesian network representing the dialogue state. As before, by using the FANTASIA tools, developers may concentrate on developing management strategies following dialogue specifications provided by Opendial to handle feedback from the remote agents.

3.5 FANTASIA tools for the interactive front-end: voice synthesis and dialogue state interaction

The first UE4 tool included in FANTASIA provides an interface towards Opendial to allow the exchange of control and feedback messages. Among the different kinds of extensions UE4 accepts, we developed an Actor Component named *OpendialConnector* to perform this task. In UE4, an Actor is an object that may be placed in the 3D world: this may be a 3D mesh, a spatialised sound, a light source, etc. An Actor Component is an object that

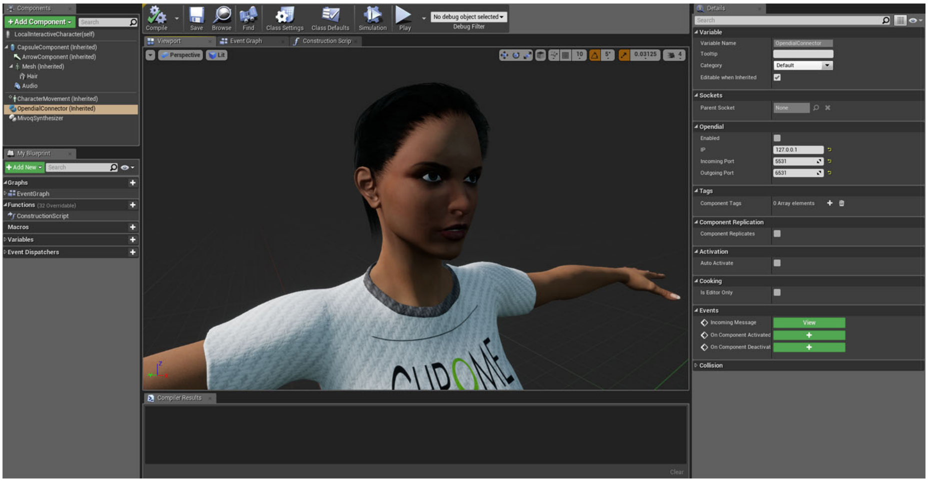

Fig. 2 A Blueprint extending the InteractiveCharacter Blueprint. The OpendialConnector tool is automatically added to the Blueprint and its configuration parameters can be set directly from the editor, making the development process consistent

may be assigned to an Actor to extend its capabilities. In the case of the OpendialConnector component, it provides access to a set of functions that allow UE4 to send and receive messages from Opendial. Leveraging on the internal UE4 event system, a custom event is used to notify Blueprint graphs when a new control message arrives. By connecting the event to the control graph, the developer is able to handle the incoming message without leaving the Blueprint editor.

Relevant tools to develop natural interactive applications are represented by synthetic voices. In the game industry, voice synthesizers are not popular as their quality is still significantly lower than the one professional actors can provide. However, synthetic voices are an important resource in research-oriented applications to allow dynamic generation of voice feedback. A modern and innovative solution in the field of synthetic voices is represented by *personalised* ones. Personalised voices are trained on a low amount of recorded speech and mimic the voice of a target person, including speaking style. This technology allows the capture and replication of specific styles such as newsreading, presenting or narrating and are of particular interest for experimental applications. In FANTASIA, we have developed a UE4 tool to integrate the Mivoq[4] speech synthesis engine [5], which provides voice personalisation and supports the standard formats for voice synthesis generation, from plain text to synthetic Speech Markup Language (SSML). To our knowledge, Mivoq is, at present, the only speech synthesis engine providing a UE4 interface accessible via Blueprint. The tool is implemented as an Actor Component named *MivoqSynthesizer*. This component exposes function calls towards audio and lipsync data getters and allows developers to select the synthesised utterance that should be played, using user defined IDs. Once the synthesis has been obtained, it is stored with the user-defined ID until a play request arrives. This means that multiple utterances can be pre-synthesised and stored, depending on the dialogue man-

[4] www.mivoq.it

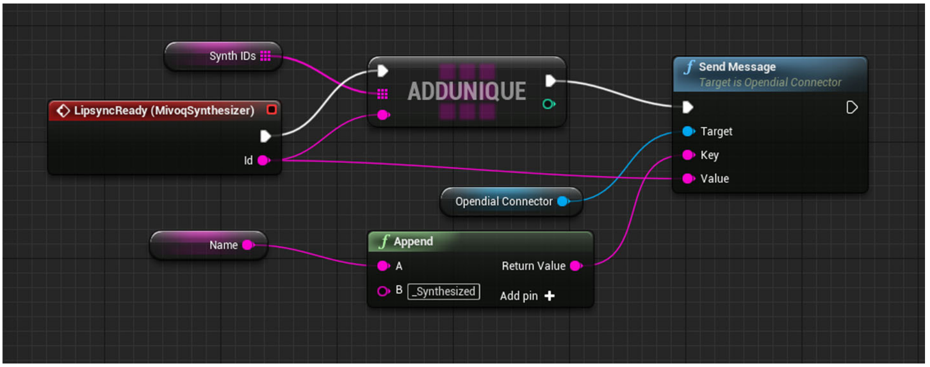

Fig. 3 An example Blueprint using the LipsyncReady event provided by the MivoqSynthesizer component and the Send Message function provided by the OpendialConnector component

ager predictions. Once a play request arrives for a stored utterance, this only has to be put in the audio component buffer associated with the MivoqSynthesizer. This allows Opendial predictive capabilities to improve the system reaction times to user requests. The MivoqSynthesizer tool exposes custom events to indicate when audio and lipsync material has been received to let developers design new graphs to handle these situations. It also exposes intensity and lipsync data to Blueprints in real time while playing voice synthesis. These data can then be used to drive any parameter linked to Blueprint nodes' properties, such as materials' physical properties or 3D mesh morphs and animations.

Since UE4 allows its basic classes to be extended by custom Blueprints, to further simplify the use of FANTASIA tools in UE4, we developed a customised Blueprint class, named *InteractiveCharacter*, which extends the default UE4 Character class. The Character class inherits from the general Actor class and represents Actors that can move around the 3D world. This information is used by the engine, for example, to optimise lighting and physics computations and improve performance at run-time. By extending the Character class, we provide developers with Blueprints that are already set up to use the FANTASIA tools for UE4. Extending the Interactive Character class by means of the Blueprints programming language, developers can immediately use the features exposed by FANTASIA for their own work. As configuration parameters are also exposed to the Blueprint editor, developers can set up the connection between UE4 and Opendial both manually and programmatically without leaving the Blueprint editor, as shown in Fig. 2.

4 Application development with FANTASIA: an example

We now provide a simple application to exemplify how the development of speech-based applications can be supported by FANTASIA. This represents only the basic use of FANTASIA as the possibilities offered by the framework go beyond this kind of application, as shown in the next section. First, we select the application domain. Following the teaching examples used to present Neo4j, we consider the movies domain. The example Movie graph contains relationships between people involved in the movie industry and the movies themselves. Suppose we want to build an application to know a single movie directed by a known person. The Opendial dialogue domain is defined in XML as follows:

```
<domain>

<settings>
<modules>opendial.modules.AgentConnector,
opendial.modules.Neo4jConnector</modules>
</settings>

<model trigger="u_u">
<rule>
<case>
<condition>
<if var="u_u"
value="Tell me the title of a movie directed by {X}." />
</condition>

<effect prob="1">
<set var="neo4j_query"
value="MATCH(n |name: '{X}'|)-[:DIRECTED]->(m)
RETURN m.title
LIMIT 1"/>
</effect>
</case>
</model>

<model trigger="neo4j_answer">
<rule>
<case>

<condition>
<if var="neo4j_answer"
value="... {X} ..." />
</condition>

<effect prob="1">
<set var="remoteAgent_outmessage"
value="Synthesize({X}, 1)"/>
</effect>
</case>

</model>

<model trigger="remoteAgent_Synthesized">
<rule>
<case>
<effect prob="1">
<set var="remoteAgent_outmessage"
value="Play({remoteAgent_Synthesized})"/>
</effect>
</case>
</model>
</domain>
```

For the sake of simplicity, in this example the system is able to respond only to a question specifically formulated in the form of *Tell me the title of a movie directed by <name>*. The *settings* section informs Opendial that the FANTASIA tools to manage connections with Neo4j and with remote actors should be loaded. The first model triggers when the variable containing user input, conventionally indicated as u_u (user utterance) in Opendial, is updated. When this happens, the input is parsed to extract the name of the person, referenced as {X}. The unification system provided by Opendial is then used to create a simple Cypher query that is put in the special variable *neo4j_query*. Note that the curly brackets required by Cypher syntax have been changed with pipe symbols: this is because curly brackets are interpreted as variable names by Opendial. The Neo4jConnector tool manages the substitution internally to produce the correct query. The Neo4jConnector tool monitors this variable and automatically submits the query to Neo4j, putting the JSON outcome in the special variable *neo4j_answer* as soon as it arrives. The second Opendial model is used to build the answer, which is sent through the AgentConnector to UE4 (connection parameters are managed separately in a different configuration file) to request the synthesis of the answer. Note that the Neo4j answer is provided in JSON format. The JSON string parsing line is shortened in this example. The request is managed to obtain audio and lipsync data from the Mivoq engine by the InteractiveCharacter Blueprint linked to the AgentConnector in Opendial. To proceed with our example, we skip the details of this part and move to the further cross-module communication step.

Let's now consider the Blueprint graph shown in Fig. 3. In the provided example, the LipsyncReady event, provided by the MivoqSynthesizer component, fires when a request for lipsync data has been managed by the synthesis engine. While the actual data are stored automatically by the component, the UE4 engine is instructed to add the ID of the synthesized material in an array of IDs, to keep track of what is available. Next, the Send Message function, provided by the OpendialConnector component, is invoked: its two parameters hold the key-value pair representing a variable update in Opendial. In this case, the graph composes the name of the variable by combining the name of the Actor to which the FANTASIA components have been associated and a user-defined suffix. The value is, of course, the ID of the synthesis request that has just been fulfilled. The OpendialConnector component sends this data to Opendial and the AgentConnector, on the other side, updates the dialogue state accordingly. This triggers the third model in Opendial, which simply instructs UE4 to play the just synthesized audio. This mechanism allows Opendial to use its predictive capabilities to request pre-syntheses to UE4 InteractiveCharacters and only play the most appropriate one when necessary. This may help to reduce the system's reaction time.

This example shows how it is possible, using FANTASIA tools, to build a cross-module system designed to query domain knowledge consistently with user requests, keep an abstract representation of the dialogue state and instruct the social front-end to provide answers. This is accomplished without leaving the high-level development workflow defined by Opendial, for the dialogue management part, and UE4, for the interface management part, thus avoiding dealing with low-level technical issues and concentrating on interaction management. This is profoundly different from the approach offered by, for example, Smartbody, which can control UE4 characters but does not offer a direct integration with the functionalities of the game engine. We would like to underline that, in the development process, the integration provided by FANTASIA in the high-level languages

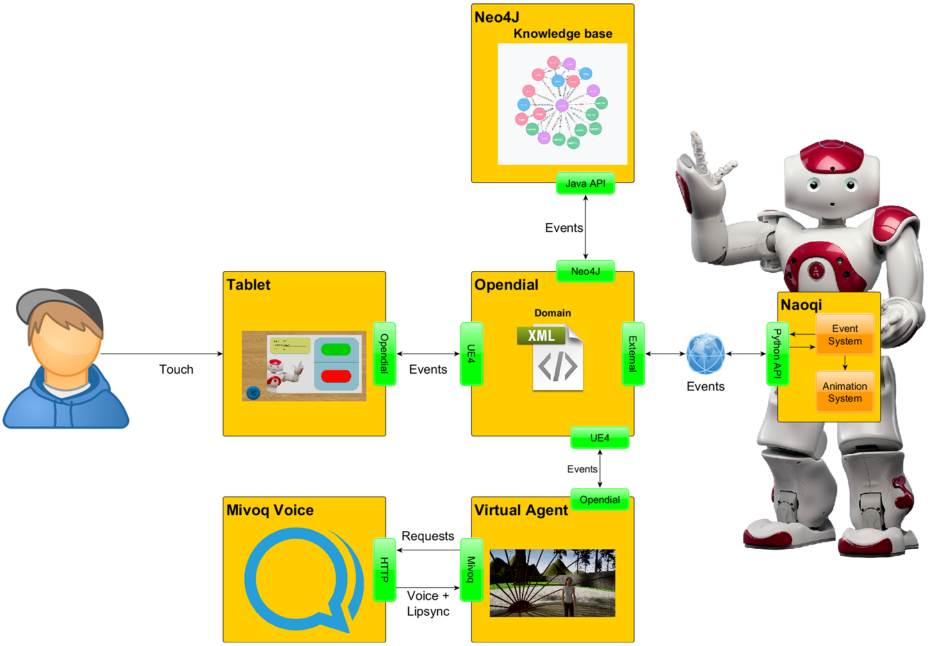

Fig. 4 System architecture for the gamified discrimination tests scenario

offered by the selected modules allows to build advanced behaviours that are not directly provided by FANTASIA without resorting to low level programming. Also, this shows how FANTASIA tools support high-level interaction design and programming throughout the different areas of knowledge representation, dialogue management and front-end presentation, whereas chatbot development frameworks concentrate on dialogue management only, leaving integration with the other areas to be developed using low-level programming.

5 Case studies

In this section, we describe how FANTASIA tools have been used to develop different kinds of architectures used for research purposes. We describe the basic applications and comment on the development process, highlighting the parts where FANTASIA was important to reduce its complexity. The first case study presents a situation where the probabilistic reasoning capabilities provided by Opendial can be used to estimate the best course of action to proceed with interaction, combined with a social front-end managed by UE4. The second case study shows how the environment provided by Opendial for Wizard-of-Oz (WoZ) experiments can be used to control a virtual reality setting in UE4 without the need to develop specific code, as this is provided by hardware producers. The benefits of being able to connect virtual reality visors with a WoZ dialogue manager with no effort are discussed. The third case study presents ongoing work on the development of virtual guides for cultural heritage, which also contains further tools that will be integrated in FANTASIA.

5.1 Case study 1: gamified discrimination tests

This case study presents a situation where FANTASIA was used to control two different agents enacting a social situation to mask a dynamically generated test for speech therapy. The goal of this case study is to demonstrate how domain knowledge representation and probabilistic reasoning support an interactive process that needs to be adjusted as user data are collected.

Phonetic perception abilities are already in place and active in the foetus, and their integrity is necessary for a normal functioning future speech development [15, 37]. Generally speaking, phonemic or phonological discrimination means that the processes of categorical perception through which differences unfold along a physical continuum (of frequency, intensity, duration) are traced to discrete categories. Since the ability to discriminate linguistic sounds is associated with the correct acquisition and production of the same sounds, an alteration of this ability could contribute to the onset of speech and language disorders [33]. At present, the question of how perception and linguistic production are interrelated is still unresolved, but there is ample evidence that highlights the relationship between production and perception in the child's phonological development, and a large number of studies show that children have a specific difficulty in discriminating the same contrasts that neutralise in their productions [2]. For this reason, evaluation of phonetic discrimination ability is important in order to individuate at-risk subjects, allowing clinicians and caregivers to operate in focused and specific ways. For preschool children (from 4 years old onward), the speech perception paradigms of identification and discrimination are the same as used by adults [27]. Among several types of discrimination test, we chose the standard AX or "same-different" procedure. Traditionally, AX tests to evaluate the phonemes discrimination capability of young children are designed as scripts and software traditionally used to administer this kind of test also follows scripts (e.g. [1]). These contain a series of (non-) word pairs presenting phoneme oppositions (i.e. *'pepi* / *'pemi*) in different syllabic structures (i.e. CV-CV is a disyllabic structure where each syllable has a single heading consonant). The child is given the task of indicating, after listening to the experimenter reading the stimuli, whether the two (non-)words are the same or different. Our test is designed in such a way that consonants presenting a single distinctive trait are opposed at each time (e.g. voiced/unvoiced, sonorant/non-sonorant). Control stimuli are present in such tests as pairs composed by the same word repeated twice and by pairs composed by completely different words. This approach is necessary as it is impossible for a human expert to dynamically select word pairs that comply with a set of very strict constraints. Specifically, each word pair must present:

- opposing consonants that differ in exactly one trait;
- the same syllabic structure in the two (non-)words;
- an opposition located in a precise position in the syllabic structure (e.g. the head consonant of the second syllable);
- stress located in the same place in the two (non-)words.

In this first case study, we represent a phonetic discrimination test as a dialogue model where each stimulus, once paired with the child's answer, generates a new stimulus as a system response. This stimulus is selected depending on a utility function taking into account both the linguistic knowledge of typical speech development [40] and the child's performance. The architecture presented in this case study was presented in [20] while the results of the experiments were presented in [22]. The architecture resulting from the deployment of FANTASIA in this setting is shown in Fig. 4.

5.1.1 Knowledge domain

In this task, we use the MultiWordNet-Extended (MWN-E) dataset [21], as the knowledge base to control the decision process for the discrimination test. The MWN-E dataset is based on the MultiWordNet dataset [26] and extended by introducing morpho-syntactic data (e.g. gender, number…), derived forms (e.g. plurals, conjugations…) and SAMPA pronunciations. Also, phonological neighbourhoods are computed and are of particular interest for this work. A word *A* is defined to be a phonological neighbour of the word *B* if it is possible to obtain *B* by altering the phonological representation of *A* using exactly one Insertion/Deletion/Substitution operation. Phonological neighbourhoods are represented by establishing HAS_PHONOLOGICAL_NEIGHBOUR relationships between two words if the Minimum Edit Distance of their phonological transcriptions equals 1. This kind of relationship has a *distance* property that, in these cases, is set to 1. HAS_PHONOLOGICAL_NEIGHBOUR relationships are also established between words that have the same pronunciation but different written forms. In this case the value of the *distance* property is set to 0. Other than the data included in the version presented in [21], the MWN-E version used in this work also contains frequency data for the terms in the vocabulary presented in the *Primo Vocabolario del Bambino* (PVB) [3] (Italian version of the MacArthur-Bates Communicative Developmental Inventories - CDI[5]) and from the Italian Wikipedia[6]. Currently, MWN-E consists of 292282 nodes containing 1536550 properties.943174 relationships among these nodes are found, phonological neighbourhood relationships at distance 1 representing the majority.

For the task of finding phonological neighbours presenting specific phones in opposition given a syllabic structure, it is possible to exploit the MWN-E database. Since the phonological transcriptions are included among the properties of words, it is possible to extract phonological neighbours at distance 1 that share the same syllabic structure to isolate phonological neighbours obtained through a substitution operation. Also, it is possible to specify which phonemes should be involved in the substitution in order to obtain the stimuli needed for the test, as shown in the following query.

```
MATCH (n)-[:HAS_PHONOLOGICAL_NEIGHBOUR
{distance: '1'}]-(m)
WHERE replace(n.phones, '1', '') =~ '^(p|b|...|LL)
(a|e|i|o|u|E|O) - (p) (a|e|i|o|u|E|O)$'
AND replace(m.phones, '1', '') =~ '^(p|b|...|LL)
(a|e|i|o|u|E|O) - (b) (a|e|i|o|u|E|O)$'
AND NOT(n.word IN(['cubo', 'cupo']))
AND NOT(m.word IN(['cubo', 'cupo']))
AND n.swearing='False' AND m.swearing='False'
RETURN DISTINCT n.word AS n, m.word AS m,
CASE WHEN n.LEt > 0 AND m.LEt > 0 THEN 2
WHEN n.LEt > 0 OR m.LEt > 0 THEN 1 ELSE 0 END AS Weight,
(n.LEt + m.LEt)/2 AS PVBMean,
(n.WkFreq + m.WkFreq)/2 AS WkMean
ORDER BY Weight DESC, PVBMean DESC, WkMean DESC
```

[5] www.sc.sdus.edu

[6] Data extracted from the 20/04/2017 Wikipedia.it dump.

This query presents the template to obtain a pair (w_1, w_2) consisting of dysyllabic words that are phonological neighbours and are obtained by substituting the /p/ phoneme in the first word with the /b/ phoneme in the second word. Sets of words to be excluded after having been presented are also included in the query (in this example, *cubo* and *cupo*). Since we work with young children, swearing words are excluded in the query by checking the corresponding attribute. The resulting words list is ordered by considering the average words frequencies. Word pairs that include words present in the PVB are given precedence. As an alternative, word pairs with the highest average Wikipedia frequency are selected. The list of consonantic phonemes is shortened for reasons of space.

5.1.2 Task management

A word pair represents a stimulus in the test. For each possible phoneme opposition, the system checks whether a stimulus with the considered syllabic structure exists. If this is the case, the stimulus is a candidate to be presented during the test and its utility is computed, at each turn, using a utility function. In this section, we summarise the principles of the statistical modelling technique used to dynamically choose the best stimulus, given the subject's observed performance among the ones obtained by querying the MWN-E database. First, on the basis of the study presented in [40], we consider the acquisition age of each phoneme. For our experiments, we refer to [29] for the distinctive features of standard Italian. The probability of a subject to assign a label to the presented opposition is a binomial distribution (Equal/Different). Therefore, to represent a priori probabilities built using previous feedback, the conjugate prior of the binomial distribution, the Beta distribution, is used. Following the Opendial implementation, a two-dimensional Dirichlet probability density function with parameters is used to model the conjugate prior. An entropy-based utility function for a given opposition is computed, assigning higher utility values to stimuli presenting the opposing features for which the associated probability density functions present the higher uncertainty.

An opposition presenting more than one highly entropic feature is not an optimal choice as it is not possible to evaluate which feature influenced the outcome. This is the reason why, for scripted tests, it is not possible to use phoneme pairs opposing more than one feature, which becomes a problem in tests opposing words as there may not be phonological neighbours with the specified structure opposing exactly the phonemes involved in the feature of interest. In a dynamically generated test, however, the estimated incapability of the subject to perceive oppositions in a given feature may allow investigation of other features that are never opposed in isolation, as in the case of the Posterior feature always being opposed with the Anterior feature. For this reason, we compute a utility function based on the mean probability value that each feature has not to be discriminated. This function assigns a higher utility value to oppositions presenting a single, highly entropic feature together with features that have been found not to be discriminated by the subject. The higher the likelihood of other features not to be discriminated, the higher the utility. Since the task complexity can be influenced by the age acquisition difference in the involved phonemes, we model a substitution-based utility function assigning a higher value to phoneme oppositions that are closer to each other in the acquisition sequence. As this is a relative measure of phoneme-based complexity for the opposition, we also need an absolute measure to prefer phonemes acquired earlier. We therefore define an acquisition-based utility function. Since all utility functions are different measures of the same object (the phoneme opposition) sharing the same range [0, 1], the final utility function for the opposition is computed as the harmonic mean of these four measures. This function lets the dialogue manager select the

Fig. 5 The experimental setup of the linguistic discrimination test

optimal stimulus for the next turn. The algorithm for dialogue management, implemented in Opendial and exploiting the MWN-E data, proposes a stimulus at each step and updates the probability distributions according to the feedback given by the subject using Bayesian inference.

The Bayesian network, in this task, represents a model of the child in terms of the probabilities that she will be able to perceive oppositions in each distinctive trait. Through the FANTASIA connection manager, Opendial is able to specify the (non-)word couples that should be presented to the child and receives the corresponding feedback. After the network update step, Opendial uses the FANTASIA Neo4j tool to query the MWN-E database and extract relevant word pairs to continue with the experiment.

5.1.3 Interaction design

The interface proposed to the child to mask the discrimination test supports a narrative in which the Nao robot wants to learn how to speak and a 3D character, named Ellie, needs the child's help to teach it. A three-polar setup, shown in Fig. 5, is established to involve the child in a socially engaging situation. Through this learning-by-teaching approach, the child is given an authoritative role to avoid making him feel threatened or evaluated. When the system starts, an introductory scenario is presented and Ellie, shown in Fig. 6, introduces herself. The scenario ends with Ellie asking the child to caress Nao in order to wake it up. This has the goal of both providing the invitation to play and establishing physical contact between Nao and the child. Whether the physical attributes of robots constitute an advantage for acceptability *per se* is still a debated issue.

In our work, we attempt to fully exploit the physical presence of the robot by presenting tasks that require the child to physically interact with it. By proposing activities that a 3D character simply cannot be involved in, we attempt to capitalise on the robot's potential to provide a more engaging multisensorial experience. Caressing, in particular, is a powerful social means to build attachment. On the other hand, the high level of control over the 3D character's movements allows efficient representation of its higher competence in the considered setup: differently from Nao, this avatar can move its lips and change its facial expressions, providing effective indications as to how to continue playing. An advantage of the presented architecture is that different virtual agents can be combined to build the test upon the various advantages they offer. After a tutorial session

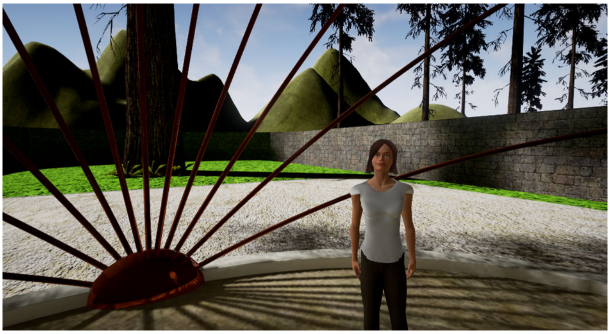

Fig. 6 The 3D character. It guides the child through the game and interacts with Nao during cutscenes

where Nao performs a small set of funny behaviours, the child is introduced to the actual test. The dialogue manager selects the most appropriate stimulus and coordinates the two agents so that Ellie, implemented using the FANTASIA InteractiveCharacter, presents the first (non-)word and Nao presents the second. The child is given one possibility to listen to the stimulus again and is required to provide a same/different feedback using an evaluation card that appears on the tablet. The interface to provide feedback is shown in Fig. 7.

The tablet interface is implemented in UE4 and controlled using the FANTASIA OpendialConnector component only. Concerning the Nao robot, we used the provided visual programming language, Choregraphe, to replicate the way UE4 manages Opendial instructions. Therefore, Opendial sends the same abstract control instructions, through the FANTASIA connection manager, towards both agents, leaving the actual implementation to the specific platform and separating the high-level experiment management logic from the realisation details.

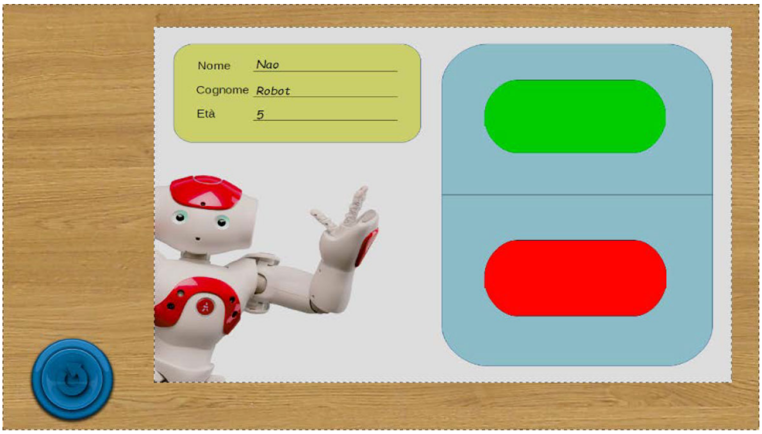

Fig. 7 The tablet interface. The child gives feedback by touching the red or green areas to evaluate Nao's performance. A repeat button is also present to allow the child to listen to the opposed words again. This is allowed only once for each stimulus

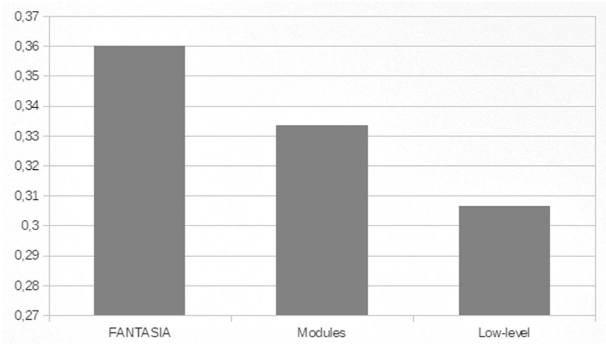

Fig. 8 Code percentages attributed to the different parts of the speech therapy application

5.1.4 Development process

This application involves different actors at the same time and needs to maximise a utility function linked to the informative power estimated for each available stimulus. In this situation, the abstract representation of the state, provided by Opendial, is crucial to concentrate the management part into a single environment, thus avoiding fragmentation in the system. To design the utility function logic, an ad hoc Java plugin was developed for this specific task. Also, the XML dialogue management specification was produced on the basis of the variables describing the children's capabilities and the other actors' statuses. Probabilistic reasoning to estimate the best stimuli to propose was automatically managed by the Opendial infrastructure on the basis of the XML dialogue domain specification. The XML dialogue domain relied on variable updates sent by other actors through the FANTASIA tools so that the need to implement communication among actors was avoided. Abstract instructions provided by Opendial through FANTASIA were delivered to the Nao robot with limited integration effort to develop the instruction parser in Choregraphe.

In UE4, the control logic for the avatar and the tablet was entirely developed via Blueprints, building the abstract representation needed by Opendial to provide instructions through the use of the OpendialConnector tool. The MivoqSynthesizer tool was directly connected to the instructions provided by Opendial, which contained the stimuli to be synthesized, and to the UE4 audio component to play the obtained synthesis. No additional programming effort was needed in UE4.

Summarising, the main programming effort needed for this task concentrated on the implementation of task-specific functions and did not involve the connection between actors or the dialogue management strategy, both of which were described either via Blueprints or via XML. Figure 8 shows the code percentages managed through FANTASIA tools, high-level languages provided by Opendial and UE4 and low-level code specifically developed for this application. Graphs report percentages over the total number of lines of code developed for this application. In the case of Blueprints, a single node has been counted as a line of code. FANTASIA tools cover most of the code as they manage communication among the different modules, MivoqSynthesizer being the most complex tool involved. Dialogue management, character animations and UI behaviour are controlled using either the Opendial XML language for dialogue specification or UE4 Blueprints. Low-level code consists of Java code developed to manage the utility computation for the available phonetic stimuli.

5.2 Case study 2: spoken commands collection for Human-Robot interaction

This case study shows how to exploit the capabilities offered by Opendial to set up Wizard of Oz experiments using 3D scenarios and characters controlled in UE4. Specifically, we describe the corpus collection procedure adopted to document human behaviour during a cooking task with a virtual agent simulating a humanoid robot in a kitchen. Given the presence of the wizard, for this case study we describe the interaction design part only. In general, however, it is also possible to use this configuration to train the Bayesian network of a dialogue domain using Wizard of Oz dialogues, as this is a functionality provided by the Opendial framework. In this case, however, we are only interested in data collection to obtain a reference corpus for natural language understanding in the cooking domain. The collected corpus represents the evaluation dataset for the EVALITA 2018 challenge Spoken Utterances Guiding Chef's Assistant Robots[7] (SUGAR).

5.2.1 Interaction design

In order to develop natural interactive systems, it is necessary, before designing the artificial intelligence to support the task, to investigate how humans would interact with the agent of interest. Wizard of Oz approaches are a common choice for this task. In the case of Human-Robot interaction, however, this is not always easy to do as the physical limitations of the machines and the technical difficulties of deploying a remotely controlled robot are many. For this reason, we created a virtual environment, depicting a kitchen, and instanced a FANTASIA InteractiveCharacter using the default mannequin mesh provided by UE4 to represent the cooking robot. During the task, participants are asked to guide the robot, named Bastian, through the steps of three recipes using voice commands. In order to let participants know about the action that Bastian should perform, we introduced a television in the kitchen: on the television screen, at each step, the system projects a movie fragment depicting a human cook while he performs the requested action. To avoid influencing participants towards the use of specific terms, the audio of the movies has been removed. Figure 9 shows an image of the virtual kitchen: Bastian stays near the kitchen island until the first command of the recipe is provided. When the simulation starts, the participant is positioned near the television to make it easier to see the movie fragments depicting the instructions. To improve the immersiveness of the session and elicit natural behaviour, the experience is delivered through virtual reality using an Oculus Rift visor. Figure 10 shows one of the participants to the corpus collection campaign while performing the task. Although Opendial can run remote instances to provide Wizard of Oz setups, using VR, this is not necessary as the wizard can operate on the same machine on which the experience is being delivered without being seen, thus simplifying the experimental setup.

As participants provide instructions to Bastian, the wizard uses the Opendial interface to

- accept the utterance: when the utterance is coherent with the presented step, Bastian executes the command and the system moves to the next one;
- ask for a repetition: when the utterance contains false starts, replanning or other production-related problems, Bastian asks the user to repeat the command;
- reject the command: when the participant did not understand the action presented on the virtual TV, Bastian states that the requested action is not correct. If the user cannot

[7] https://sites.google.com/view/sugar-evalita/

Fig. 9 The virtual kitchen used for the data collection task. Bastian stays near the kitchen island until the first command is issued. He then moves near the stove to perform the cooking animations. The television on the wall presents video fragments of human cooks performing the steps of the current recipe

Fig. 10 One of the participants in the Kitchen Commands corpus collection campaign. Using VR, we both provide an immersive situation to elicit natural behaviour in giving commands to the virtual robot and efficiently hide the wizard, on the back, who is monitoring the virtual experience and controlling Bastian using the Opendial interface

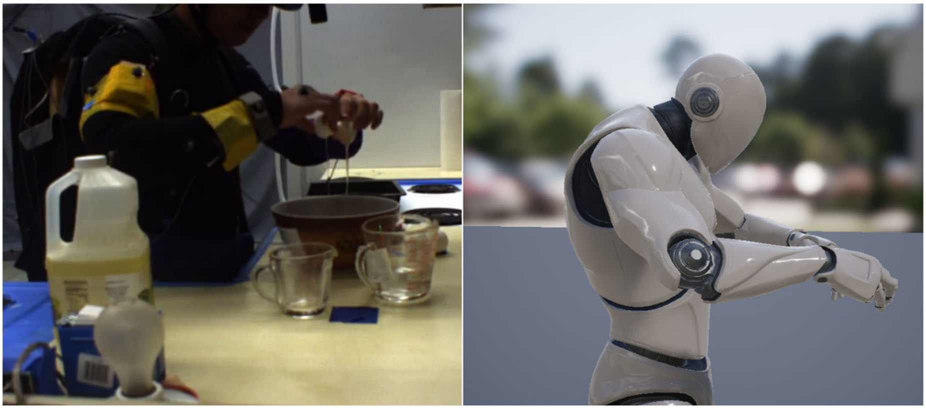

Fig. 11 A frame of the eggs breaking animation performed by Bastian (on the right) and the corresponding movement from the original actor in the CMU database (on the left)

make sense of the step three times, the wizard lets the session continue with the next step and the utterance is discarded.

Bastian's animations for generic dialogue interactions were selected among freely available motion capture animations for UE4. A set of cooking-related actions were also included by importing motion capture data from the CMU Multi-Modal Activity Database [35], which provides recordings of people involved in cooking and food preparation tasks. The original mocap skeleton has been retargeted to the standard humanoid skeleton provided by UE4 to let Bastian exhibit natural movements as positive feedback to the user. Figure 11 shows a frame of the original actor in the CMU database and the corresponding movement performed by Bastian. To improve the quality of the experience, cooking sound effects have been used to provide further information about the status of the environment and about Bastian's actions (fire starting and ending, frying oil, eggs breaking, etc...). The architecture obtained by deploying FANTASIA tools on this task is shown in Fig. 12.

5.2.2 Development process

The main effort for this task was the modelling of the virtual environment in which Bastian was immersed and in creating its animations, for which UE4 provided adequate tools. No effort was needed to integrate the VR display in the engine as this was already provided by the Oculus developers. Also, no effort was needed to create the WoZ experiment, as Opendial supports such a scenario. The connection between Opendial and the InteractiveCharacter implementation produced to control Bastian was managed by FANTASIA while the animation logic was entirely developed in Blueprints. There was no need to produce additional code to manage this task. Figure 13 shows that FANTASIA let us take advantage of both UE4 and Opendial characteristics to develop the application.

6 Case study 3: virtual guides for cultural heritage

This case study demonstrates how FANTASIA can be used to let users access a semantically rich 3D environment in the case of virtual guides development for cultural heritage. Several

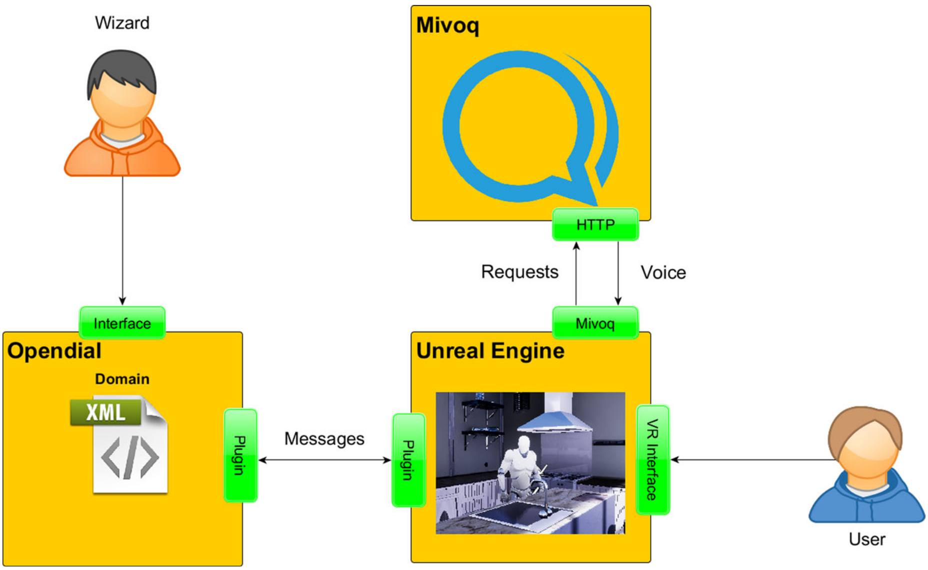

Fig. 12 Final architecture for the data collection campaign of the EVALITA 2018 challenge on spoken language understanding

studies reveal that most people visit museums in groups. However, most exhibits and digital installations are designed for single users. Therefore, the possibility of involving groups of visitors in a shared interactive experience represents a challenge for system designers in the field of cultural heritage [10] while the use of Embodied Conversational Agents in public spaces such as museums has been investigated in previous works [14, 36]. In this case study, we deploy the FANTASIA tools to build an architecture for cultural heritage content delivery to small groups of museum visitors. The application consists of a question answering system supporting multimodal commands (speech + gestures). The experimental setup consists of a large curved screen (4m x 3m) on which cultural environments are projected. This case study extends the work presented in [38] and consists of ongoing work in the framework of the Italian PRIN project Cultural Heritage Resources Orienting Multimodal Experiences (CHROME), covering the three Campanian Charterhouses.

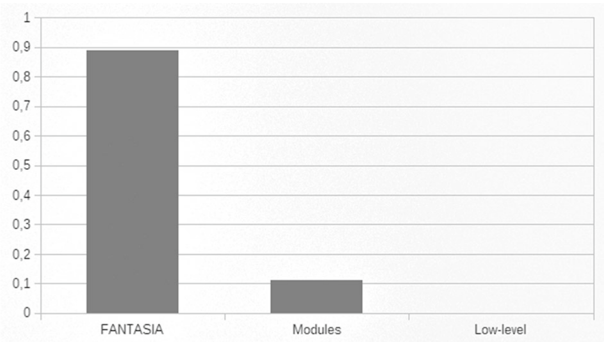

Fig. 13 Code percentages attributed to the different parts of the corpus collection scenario

6.1 Knowledge representation

For this task, we use domain knowledge about art and architecture coming from the Getty's Institute vocabularies. These resources contain structured terminology for art, architecture, decorative arts, archival materials, visual surrogates, conservation and bibliographic materials. They are distributed in Linked Open Data format and allow collaborative extensions by verified data providers. In our specific case study, we mainly consider the Art and Architecture Thesaurus (AAT) [25]. This particular resource provides terms, descriptions and other metadata for generic concepts related to art, architecture, conservation and archaeology. Information on work types, styles, materials and techniques is also included. Data are provided in RDF format following W3C guidelines. In order to use the AAT data, the graphs that are relevant in the objects of interest for the CHROME project, the Campanian Charterhouses, are being transferred in the Neo4j database. Using only the subgraphs of interest, in this case, is necessary as the foreseen interactive applications are intended for Italian users: as the AAT languages do not include Italian, the translation effort has been limited to the relevant concepts for reasons of efficiency. In line of principle, however, all the Getty resources can be imported in Neo4j and consequently used in the proposed architecture.

In order to let the system access textual knowledge related to the relevant concepts, during the CHROME project we collected a corpus of different kinds of text describing the Campanian Charterhouses. Following the approach described in [23], these texts are being organised in tree-like structures describing deepening levels of information and shared topics. These structures can be efficiently represented and traversed using Neo4j and are, therefore, part of the domain knowledge represented in the database.

Differently from the approach presented in [38], where objects of interest, paintings, were clearly separated from each other, in this case study we consider complex objects, architectural 3D reconstructions. In this situation, a more powerful representation of spatial knowledge is needed, as multiple concepts, often blending into each other, are found on a single 3D model. For this reason, we introduced an original method to annotate 3D models with semantic data in UE4 based on the use of available tools that are normally used for texturing. While the module supporting this method for spatial knowledge representation is not yet part of FANTASIA, it will be included in the toolset as soon as integration with Blueprints is completed. More details about 3D model annotations with semantic maps can be found in [4].

The annotation consists in using maps describing semantic concepts applied to the 3D model like a texture, thus avoiding the need to geometrically segment the architectural artefact. For each concept in the AAT found in the digital architectural model, a semantic map is created and assigned the same unique ID as the concept it represents is recorded with in the AAT. As an improvement with respect to previous approaches, the semantic information is represented as a grayscale map: each map records which polygons, in the digital model, are relevant for the concept it represents by using the model's UV map. In our approach, white indicates high relevance, while black indicates no relevance. An example of a semantic map is shown in Fig. 14. Using semantic maps and reference IDs for the annotated concepts allows the integration of multiple sources of information (texts, images, audio recordings, etc...) sharing the same annotation scheme. Cross-referencing these sources opens the possibility of producing advanced interfaces to link the descriptions a specific artefact has in separate domains.

The possibility of using gradients in the map lets annotators refine the quality of the semantic data. This way, it is possible to express, more than a binary relevance of each

Fig. 14 A semantic map for the *pediment* concept. The 3D model represents a part of the monumental staircase in the San Lorenzo Charterhouse (Padula, Italy)

polygon for a given concept, a relevance *level* for that concept. This is important in the field of architectural heritage, as it is not always possible to classify an element in a unique and precise way, and it becomes useful when an architectural element cannot be assigned to a specific category. The same applies to situations in which it is not possible to indicate where, exactly, an architectural element becomes another one. This also makes it possible to consider semantic maps produced by multiple annotators to obtain a final map by computing the mean values for each UV coordinate, similarly to what has been done in other fields where annotation uncertainty is important, such as for emotions [18].

6.2 Task management

To build the multimodal question answering system, the following input recognition functions have been implemented:

- *Natural Language Understanding* (NLU): processes speech signals to obtain a semantic interpretation;
- *Pointing Recognition* (PR): recognises which objects are pointed by users;
- *Active Speaker Detection* (ASD): identifies the last active speaker.

NLU has been implemented by using a Speech Recognition Grammar Specification (SRGS) rule set generated in a semi-automatic way using the MWN-E database [21] and the approach presented in [6]. The PR and the ASD modules have been implemented using Kinect and the corresponding plugin to integrate the device in UE4. Specifically, mapping between tracked bodies and skeleton assets provided by UE4 was used to directly project users inside the 3D scene. This way, the PR algorithm is able to analyse joints' coordinates extracted by each skeleton asset to compute the pointing direction based on the shoulder-hand line. Using raycasting together with the collision detection system provided by UE4, collision points with actors in the environment are used to determine what the users are pointing at. While in [38] it was sufficient to detect which object raised the collision event, in the case of complex 3D architectural models the relevant information is found in the semantic maps associated with the model itself. By providing Opendial with the concept codes that are relevant for the incoming query using the FANTASIA tools, the dialogue manager can access the knowledge included in Neo4j to extract the relevant information.

The ASD module, based on a similar strategy, combines the sound source angle provided by Kinect with skeletal assets locations to determine the probabilities that each detected user has to be the active one over time. Active speaker data are also passed to Opendial using the FANTASIA tools. The dialogue manager maintains the dialogue state to allow the user to submit contextualised queries. For example, after asking about the author of a specific painting (e.g. "When was that artwork created?"), the system is able to provide an answer to further queries by considering the current context, as modelled by the Bayesian network (e.g. "And that one?"). Probabilistic reasoning is important, in this case, as input is noisy both for speech and for gestures, as demonstrated in [38]. Also, to implement more complex behaviour in the future, estimating users' interests and adapting the presentation accordingly is an important feature to introduce.

In the extension we are developing for the original task, having FANTASIA link UE4 to Opendial allows spatial knowledge to be cross-referenced with domain knowledge on an abstract level. It also allows the system to take advantage of the probabilistic framework on which Opendial is based to handle possible uncertainties concerning the displayed environments. Specifically, the information encoded in the semantic maps can be used to evaluate what users are asking information about by considering the potential noise inherent in the Kinect tracking system but also implicit ambiguities found in the architectural artefact.

6.3 Interaction design

For the task presented in [38], a 3D scene was designed in UE4, showing three slots in which a set of artworks belonging to one of the available categories (Impressionism, Renaissance, etc…) is dynamically placed according to the users' requests. Users can ask to change the displayed category and, according to a finite set of verbal action classes, they can obtain information about a specific painting shown in the 3D environment. Users are free to refer to artworks by using pointing gestures and/or deictic and spatial expressions. The interface projected on the screen during the experience is shown in Fig. 15. Results presented in [38] showed that people were generally interested in the system and were able to use it to obtain information.

In the updated version of the system, 3D environments obtained using terrestrial laser scanning and photogrammetric reconstruction techniques are projected on the screen and the same multimodal strategies are available to submit queries to the system. In this first version of the system, we used a non-anthropomorphic conversational agent, in the form of a lighted globe, to provide answers. Since the MivoqSynthesizer component also exposes intensity data in real time, once the Play function is invoked, connecting these data to the globe material *Emissive* property was straightforward using the UE4 Material Blueprints. This allows the globe to *blink* consistently with the audio that is generated on the fly, creating an effect that is easy to set up in the Blueprint editor and has multiple applications.

While in the work presented in [38] we used a non-anthropomorphic conversational agent to provide feedback to the users, the proposed architecture will be used to control a 3D character, implemented as a FANTASIA InteractiveCharacter that will accompany the users in their exploration of the surveyed 3D environments. Semantic maps, in this case, will be used to control pointing gestures to accompany the presentation. The control system for the conversational agent will be modelled on the basis of the results of a multimodal corpus collection campaign to study how expert art historians relate themselves to the

Fig. 15 The interface projected on the large curved screen to let users inquire about paintings in [38]. A non-anthropomorphic conversational agent, modelled as a blinking globe, is used in this case

cultural environment, which contains 12 hours of audio-visual material. The corpus collection methodology is presented in [24] and is aimed at providing the social agent with a basic set of communicative strategies, both verbal and non-verbal, for virtual guides immersed in the reconstructed environments, as shown in Fig. 16. The system architecture developed for this task using the FANTASIA tools is shown in Fig. 17.

6.4 Development process

At the time of writing, the pointing system based on semantic maps and on the UE4 collision event system is undergoing the last stages of evaluation. Specialised UE4 Blueprints extending the base meshes, which we call SmartMeshes, have been developed and are provided with specialised material Blueprints, which we call SmartMaterials. SmartMaterials extend normal Materials by providing support for the additional maps. The collision system

Fig. 16 A virtual character immersed in the 3D reconstruction of an environment of the San Lorenzo Charterhouse in Padula

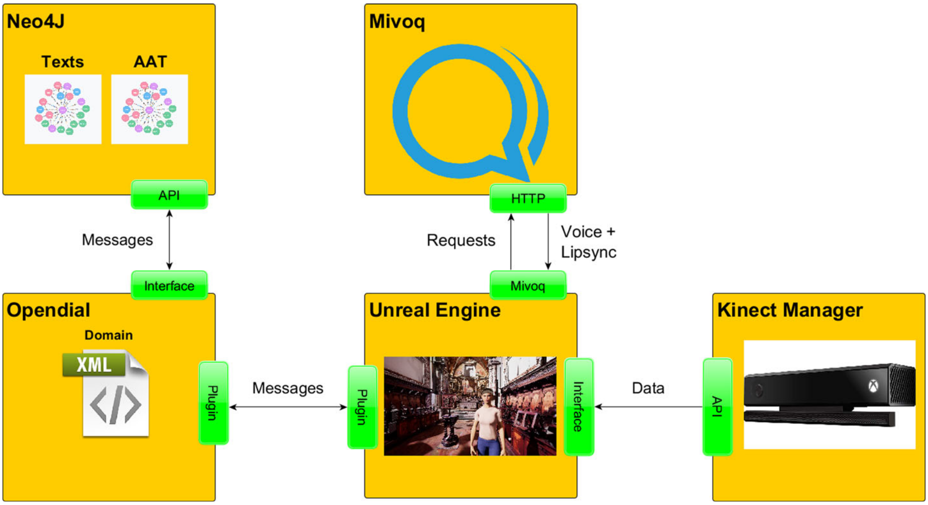

Fig. 17 The architecture used to control question answering systems for cultural heritage

included in UE4 is used to query SmartMaterials when users point towards a SmartMesh. The OpendialConnector is used to update the dialogue manager state with the semantic labels associated with the part of the SmartMaterial the user is pointing at. Preliminary results conducted in the laboratory indicate that accessing semantic maps using pointing gestures captured with Kinect is comparable to using a mouse on a desktop. An evaluation campaign in the San Martino Charterhouse is being conducted to evaluate the system in a less controlled environment.

SmartMaterials will be integrated in FANTASIA as soon as testing is complete. For this reason, they are still considered low-level code in Fig. 18. In this case, FANTASIA still manages most of the application logic and the percentage will increase once SmartMeshes and SmartMaterials are included in the toolset. The Kinect interface integrated with Blueprints avoids the need to manage access to the sensor at low level and was entirely managed in the UE4 visual programming environment.

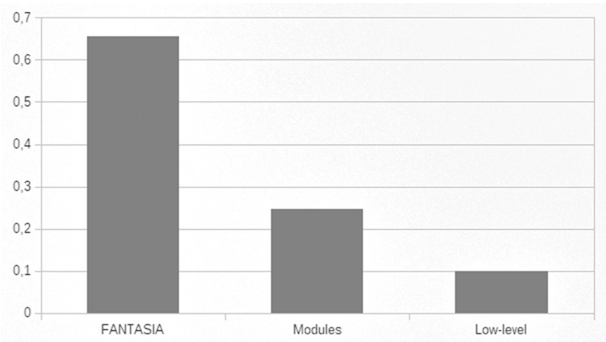

Fig. 18 Code percentages attributed to the different parts of the virtal guides scenario

7 Conclusions

We have presented FANTASIA, a toolset to make the development of interactive applications for HCI research purposes fast and coherent with the design strategies of a set of freely available modules, each specialised on particular aspects of interactive systems. FANTASIA extends the possibilities offered by UE4, a high-profile game engine, by integrating research-oriented modules for knowledge representation, dialogue management, probabilistic reasoning and voice synthesis, creating a seamless workflow for interactive systems development. FANTASIA has already been deployed in a number of different use cases. The presented case studies show how the FANTASIA tools allow the development of very different applications sharing the same technological core. FANTASIA lets developers combine the strengths of each module without leaving the basic design process that each of these tools already provides, thus making the development process consistent and structured.

With respect to other, more general, frameworks, FANTASIA is tied to specific solutions but, because of this, it takes advantage of industry grade technology and maintenance, allowing HCI researchers to concentrate on aspects that are not usually covered by these solutions, like knowledge representation and dialogue management. FANTASIA builds upon the industrial push to develop high-end game engines and on the interest device producers have to integrate their products into these engines. FANTASIA extends this by providing support to a kind of technology that, at present, is closer to the academic world. Our position is that, at present, taking advantage of the powerful tools made available by the industry is also convenient for HCI researchers from a system design point of view: using game engines for visualisation only, while they are designed to manage efficiently aspects like animation, situation monitoring and working logic, implies renouncing to efficient technology that is already integrated in them. This inevitably has an effect on performances, development speed and efficiency as the contents provided by external modules must be replicated in the engine although the engine itself would have been able to manage the externalised logic. Developing research applications with FANTASIA is, therefore, convenient for HCI researchers as they do not have to struggle to keep up with industry grade technological standards on aspects that are not closely related to their research but do, indeed, have an impact. This lets them concentrate on testing models for interaction management that rely on finer knowledge representation approaches and, typically, on probabilistic reasoning.

Although already usable to develop working applications, FANTASIA needs to be aligned to the standard formats for multimodal representation in order to support integration with tools that have been developed in the past, for example to better support SAIBA-compliant modules. Also, at the present time, FANTASIA uses a direct-link communication protocol to manage message exchanges between the dialogue manager and the remote agents. Similarly to what has been done with Smartbody, however, tools to integrate a message queuing server will be provided in the near future to simplify the development of systems composed by many remote agents.

Acknowledgements Antonio Origlia's work is funded by the Italian PRIN project *Cultural Heritage Resources Orienting Multimodal Experience* (CHROME) #B52F15000450001.

References

1. André C, Ghio A, Cavé C, Teston B (2007) PERCEVAL: a computer-driven system for experimentation on auditory and visual perception. arXiv:0705.4415
2. Byun TM, Tiede M (2017) Perception-production relations in later development of american english rhotics. PloS One 12(2):e0172,022
3. Caselli MC, Casadio P (1995) Il primo vocabolario del bambino. Franco Angeli, Milano
4. Cera V, Origlia A, Cutugno F, Campi M (2018) Semantically annotated 3d material supporting the design of natural user interfaces for architectural heritage. In: Proceedings of the AVI-CH workshop (to appear)
5. Cosi P, Paci G, Sommavilla G, Tesser F (2016) Mivoq-ptts-a revolutionary new way of thinking tts. In: Proceedings of interspeech, pp 3888–3889
6. Di Maro M, Valentino M, Riccio A, Origlia A (2017) Graph databases for designing high-performance speech recognition grammars. In: IWCS 2017—12th international conference on computational semantics—short papers
7. Dietze F, Karoff J, Valdez AC, Ziefle M, Greven C, Schroeder U (2016) An open-source object-graph-mapping framework for neo4j and scala: Renesca. In: International conference on availability, reliability, and security. Springer, pp 204–218
8. Drakopoulos G, Kanavos A, Makris C, Megalooikonomou V (2015) On converting community detection algorithms for fuzzy graphs in neo4j. In: Proceedings of the 5th International Workshop on Combinations of Intelligent Methods and Applications, CIMA
9. González J, Escobar J, Sánchez H, De la Hoz J, Beltrán J (2017) 2d and 3d virtual interactive laboratories of physics on unity platform. In: Journal of physics: conference series, vol 935. IOP Publishing, p 012069
10. Hornecker E, Stifter M (2006) Learning from interactive museum installations about interaction design for public settings. In: Proceedings of the 18th Australia conference on computer-human interaction: design: activities, artefacts and environments. ACM, pp 135–142
11. Irwansyah F, Yusuf Y, Farida I, Ramdhani M (2018) Augmented reality (ar) technology on the android operating system in chemistry learning. In: IOP Conference series: materials science and engineering, vol 288. IOP Publishing, p 012068
12. Jiménez P, Diez JV, Ordieres-Mere J (2016) Hoshin kanri visualization with neo4j. empowering leaders to operationalize lean structural networks. Procedia CIRP 55:284–289
13. Kersten TP, Tschirschwitz F, Deggim S (2017) Development of a virtual museum including a 4d presentation of building history in virtual reality. Int Archives Photogrammetry Remote Sens Spatial Inform Sci 42:361
14. Kopp S, Gesellensetter L, Krämer NC, Wachsmuth I (2005) A conversational agent as museum guide–design and evaluation of a real-world application. In: International workshop on intelligent virtual agents. Springer, pp 329–343
15. Kuhl PK (2004) Early language acquisition: cracking the speech code. Nature Rev Neuroscience 5(11):831–843
16. Lison P, Kennington C (2016) Opendial: a toolkit for developing spoken dialogue systems with probabilistic rules. ACL 2016:67
17. Martinie C, Navarre D, Palanque P, Barboni E, Canny A (2018) Toucan: an ide supporting the development of effective interactive java applications. In: Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems. ACM, p 4
18. McKeown G, Valstar MF, Cowie R, Pantic M (2010) The SEMAINE corpus of emotionally coloured character interactions. In: Proceedings of ICME, pp 1079–1084
19. Niewiadomski R, Bevacqua E, Mancini M, Pelachaud C (2009) Greta: an interactive expressive eca system. In: Proceedings of the 8th international conference on autonomous agents and multiagent systems-Volume 2, 1399–1400. International Foundation for Autonomous Agents and Multiagent Systems
20. Origlia A, Cosi P, Rodà A, Zmarich C (2017) A dialogue-based software architecture for gamified discrimination tests. In: Proceedings of the first workshop on games-human interaction @ CHItaly
21. Origlia A, Paci G, Cutugno F (2017) MWN-E: a graph database to merge morpho-syntactic and phonological data for italian. In: Proceedings of Subsidia
22. Origlia A, Rodà A, Zmarich C, Cosi P, Nigris S, Colavolpe B, Brai I (2018) Gamified discrimination tests for speech therapy applications. In: Proceedings of the annual conference of the Italian association of speech science (AISV) (to appear)
23. Origlia A, Rossi A, Chiacchio ML, Cutugno F (2016) Cultural heritage presentations with a humanoid robot using implicit feedback. In: Proceedings of the AVI-CH workshop

24. Origlia A, Savy R, Poggi I, Cutugno F, Alfano I, D'Errico F, Vincze L, Cataldo V (2018) An audiovisual corpus of guided tours in cultural sites: data collection protocols in the chrome project. In: Proceedings of the AVI-CH workshop (to appear)
25. Petersen T (1990) Developing a new thesaurus for art and architecture. Library Trends 38(4):644–658
26. Pianta E, Bentivogli L, Girardi C (2002) Developing an aligned multilingual database. In: Proceedings of the 1st international conference on global wordnet
27. Polka L, Jusczyk PW, Rvachew S (1995) Methods for studying speech perception in infants and children, Speech perception and linguistic experience: Issues in cross-language research, 49–89
28. Qiu W, Yuille A (2016) Unrealcv: connecting computer vision to unreal engine. In: European conference on computer vision. Springer, pp 909–916
29. Schmid S (1999) Fonetica e fonologia dell'italiano Paravia scriptorium
30. Shah S, Dey D, Lovett C, Kapoor A (2018) Airsim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics. Springer, pp 621–635
31. Shrinivasan YB, Zhang Y (2017) CELIO: an application development framework for interactive spaces. arXiv:1710.01772
32. Squire K, Jenkins H, Holland W, Miller H, Alice O, Philip Tan K, Todd K (2003) Design principles of next-generation digital gaming for education, Educ Technol, 17–23
33. Tallal P (1976) Rapid auditory processing in normal and disordered language development. J Speech Language Hear Res 19(3):561–571
34. Thiebaux M, Marsella S, Marshall AN, Kallmann M (2008) Smartbody: behavior realization for embodied conversational agents. In: Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1, 151–158. International Foundation for Autonomous Agents and Multiagent Systems
35. De la Torre F, Hodgins J, Bargteil A, Martin X, Macey J, Collado A, Beltran P (2008) Guide to the carnegie mellon university multimodal activity (cmu-mmac) database, Robotics Institute, 135
36. Traum D, Aggarwal P, Artstein R, Foutz S, Gerten J, Katsamanis A, Leuski A, Noren D, Swartout W (2012) Ada and grace: direct interaction with museum visitors. In: Intelligent virtual agents. Springer, pp 245–251
37. Tsao FM, Liu HM, Kuhl PK (2004) Speech perception in infancy predicts language development in the second year of life: a longitudinal study. Child Development 75(4):1067–1084
38. Valentino M, Origlia A, Cutugno F (2017) Multimodal speech and gestures fusion for small groups. In: Proceedings of the workshop on "designing, implementing and evaluating mid-air gestures and speech-based interaction" @ CHItaly 2017 [online]
39. Webber J (2012) A programmatic introduction to neo4j. In: Proceedings of the 3rd annual conference on systems, programming, and applications: software for humanity. ACM, pp 217–218
40. Zmarich C, Bonifacio S (2005) Phonetic inventories in Italian children aged 18-27 months: a longitudinal study. In: INTERSPEECH, pp 757–760

Antonio Origlia born in 1982 in Naples (Italy). He obtained his PhD in 2013 with a thesis on automatic prosodic analysis for interactive robotics with a focus on affective computing. His research interests cover natural interfaces using multimodal inputs and machine learning to manage human-computer interaction, with a focus on cultural heritage applications. He participated to 5 national and international research projects. Among them: SHERPA (Smart collaboration between Humans and ground-aErial Robots for imProving rescuing activities in Alpine environments), OR.C.HE.S.T.R.A. (ORganization of Cultural HEritage and Smart Tourism and Real-time Accessibility) and CHROME (Cultural Heritage Resources Orienting Multimodal Experiences).

Francesco Cutugno

Antonio Rodà is Assistant Professor of Computer Engineering at the Dept. of Information Engineering of the University of Padova. His main research interests are in computational models of expressive music performance, affective computing, interactive multimodal learning environments, digital preservation and restoration of musical cultural heritage. He has participated in 11 national and international research projects, among them: 2014-2015 Archaeology Virtual Acoustics. A pan flute from ancient Egypt, CPDA 133925; 2010-2012 DREAM (Digital Reworking/reappropriation of ElectroAcoustic Music); 2004-2008 ENACTIVE (Enactive Interfaces), European Network of Excellence. He is author and co-author of more the 100 paper on national and international journals and peer reviewed conferences.

Piero Cosi born in 1956 in Padua (Italy). He received the Degree on Electronic Engineering from Padua University in 1981. He is currently a speech technology researcher of ISTC CNR, UOS Padova ("Istituto di Scienze e Tecnologie della Cognizione - Unità Organizzativa di Supporto di Padova of the National Research Council") of which He is the Director/Responsible. His scientific activity has been mainly concerned with: signal processing, speech analysis, auditory modelling, automatic speech recognition, neural networks, emotional text-to-speech, facial animation, interactive talking robot. He is author or co-author of more than 200 papers published in Books, International Journals and Conferences, and he is leading, with the role of Scientific Responsible of Operative Unit, many International, European and Italian Projects. He is the Director of the "Speech and Multimodal Communication Laboratory" in which various research and development activities concerning the implementation of the Italian versions of various SOFTWARE systems are carried out: CSLU Speech Toolkit, SONIC, SPHINX, JULIUS, BAVIECA and KALDI for automatic speech recognition, FESTIVAL, MBROLA and MaryTTS for synthesis from written text and BALDINI, and LUCIA for the creation of expressive and emotional speaking virtual agents. He is the project manager of ILT (Italian Literacy Tutor) an integrated and interactive program for Italian language teaching. He is and he has been responsible for various national (TICCA, MIUR-FIRB 2003, Speaky Acutattile) and international projects (EU FP6, FP7 - MAGICSTER, PF-Star, ALIZ-E) related to speech technologies and robotics.

Claudio Zmarich is Senior Researcher at CNR- Istituto di Scienze e Tecnologie della Cognizione (ISTC), local section of Padova, and Charged Professor of Phonetics and Phonology, at the Logopedic Course, University of Padova. His research interests regard Articulatory Phonology, Speech and Language Pathology, especially preschool stuttering, and Speech acquisition, especially (co)articulatory development. Dr. Zmarich is author of more than 150 publications in international and national scientific journals, proceedings and books (abstracts excluded).

Affiliations

Antonio Origlia[1] · **Francesco Cutugno**[2] · **Antonio Rodà**[3] · **Piero Cosi**[4] · **Claudio Zmarich**[4]

Francesco Cutugno
cutugno@unina.it

Antonio Rodà
roda@dei.unipd.it

Piero Cosi
piero.cosi@cnr.it

Claudio Zmarich
claudio.zmarich@cnr.it

1 URBAN/ECO Research Center, University of Naples "Federico II", 80138 Napoli, NA, Italy

2 Department of Electrical Engineering and Information Technology, University of Naples "Federico II", 80138 Napoli, NA, Italy

3 Department of Information Engineering, University of Padua, 35122 Padova, PD, Italy

4 Institute of Cognitive Sciences and Technology (CNR-ISTC), 35137 Padova, Italy