Swoozy - An Innovative Design of a Distributed and Gesture-based Semantic Television System

Matthieu Deru and Simon Bergweiler

German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
Email: firstname.lastname@dfki.de

***Abstract*—In this article, we describe an innovative approach to an intelligent television system named *Swoozy* that enables viewers to discover extended information such as facts, images, shopping recommendations or video clips about the currently broadcasted TV program by using the power of technologies of the Semantic Web (Web 3.0). Via a gesture-based user interface viewers will get answers to questions they may ask themselves during a movie or TV report directly on their television. In most cases, these questions are related to the name and vita of the featured actor, the place where a scene was filmed, or purchasable books and items about the topic of the report the viewer is watching. Furthermore, a new interaction concept for TVs is proposed using semantic annotations called "Grabbables" that are displayed on top of the videos and that provide a semantic referencing between the videos' content and an ontological representation to access Semantic Web Services.**

Keywords–interactive television system; Semantic Web Technologies; Web 3.0; video annotation; gesture-based interaction.

I. INTRODUCTION

A study conducted by the German marketer for audiovisual media SevenOneMedia [1], reveals that in a viewer panel aged between 14 and 29, 45 % of them are surfing in parallel of watching television and that the main purpose of this browsing is to find out more information about the program, e.g., an actor's name or biography, a location or a depicted product. This search is likely done by either using a mobile or TV-app or by proactively typing in a keyword or complete phrase in a Web search engine.

The current development trend in interactive connected television systems is very app-oriented and forces users to install a lot of single apps, for example, one for searching videos another one for images. In cases when no suitable apps are found, users can interact with the TV's inbuilt Web browser to get additional information. Unfortunately the switch between several apps will oblige the user to leave his TV program and to interact several times with his remote controller before finally getting the information he was looking for.

The following approach discusses and shows a new way how viewers can interact with additional content while watching a TV program. They can search in parallel for information in the Web and easily browse through the found results without an interaction breach. In a first implementation, the developed prototype system relies on semantic annotations gained out of the analysis of a broadcasted video combined with gesture-based interactions that will enable users to directly start a search in the Web using Semantic Web technologies and get precise results in relation to the current scene like videos, text or news articles, pictures, and shopping recommendations.

Whereas systems like [2][3][4][5] are using the Semantic Web for detecting possible matches between the watched program and other Web-based contents and to only offer a personalized TV access, our approach uses semantics on several levels. The first level is the extraction of knowledge and concepts from an ordinary non pre-annotated Digital Video Broadcasting (DVB) signal, from a standard television provider. From this TV data stream, the required information is extracted and transferred by matching rules to annotations, which are necessary input to trigger a semantic search. Over an intuitive dedicated gesture-based graphical TV interface, presented in section IV, the viewer can then easily trigger a search using semantic queries. These queries are then finally processed by a specially designed and implemented engine called Joint Service Engine (JSE), which uses the Semantic Web, ontologies and semantic mappings to return context and domain sensitive results, as described in section V.

The prototype was implemented in form of setup box-based software solution to demonstrate the technically feasibility of a gesture based interactive television system combined with semantic processing, even if the current broadcasting infrastructures do not fully provide all annotations and information required for this task.

In section II, this paper gives an overview of existing and used Semantic Web technologies and shows how annotations and semantic information can be extracted after an audiovisual analysis of a TV signal. Section III presents in detail each implemented module, which is used during the extraction process. In section IV, the choices for the design of the user interface are motivated and the method how gesture interactions leads to a semantic search is presented. Section V, before the summary, will give an insight view on how the Semantic Web is used to query and deliver enriched multimedia results to the viewer.

II. RELATED WORK

A. Semantic Web technologies

The power of the Semantic Web (Web 3.0) [6] with its technologies resides in the fact that several information sources on the Web can be used in different combinations to establish new relations between conventional semantic representations of knowledge, such as ontologies, Resource Description Framework (RDF) triple stores [7], and common Web service interfaces in form of service mashups [8].

The World Wide Web Consortium (W3C) has declared ontologies as an open standard for describing information of

 ISBN: 978-1-61208-353-7

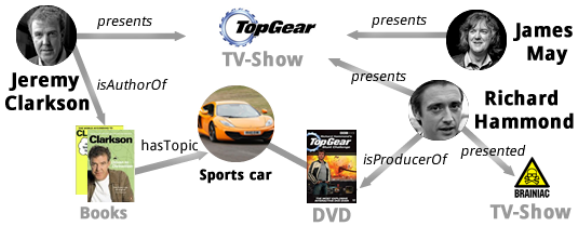

Figure 1. Discovering new semantic relations in a TV domain

an application domain and also defined appropriate ontological description languages such as RDF(S) [7][9] and OWL [10]. Ontologies, as specification languages have been specially developed for use in the Semantic Web and consists of concepts and relations. Relations organize concepts hierarchically and put them together in any relationship. These relations provide a quick access to important information in a given domain, like the biography of a presenter or speaker, interesting books or shopping items. Figure 1 shows an example of how those relations can be used to find out more information about the TV program TopGear. Starting from the TV show the three main characters, Jeremy Clarkson, Richard Hammond, and James May can be found, with further references to written books or produced DVDs. A further conclusion based on all of these relations leads to a science show named Brainiac that was also presented by Richard Hammond a few years ago.

But, in order to give viewers the access to these new relations and their contents, a relation between the video's content and its semantic representation must be established: the viewed video must be annotated or better said a mapping between what the viewer is currently seeing (e.g., *a person is speaking*) and the full scene description (e.g., *this person is a politician named Barack Obama, he is the President of the United States and is giving a speech*) along with semantic annotations must be achieved through semantic mapping. This mapping combines visual information from the current scene and ontological concepts like (person, fictional character, object, and monument). Through this assignment, extracted domain knowledge is classified [11]. This gain of knowledge out of a video can only be realized by video-based annotations: in our system we call these semantic terms.

Although several tools [12][13] and solutions exist for embedding metadata and annotations into a video - most of them are working with XML-based annotation formats like Broadcast Metadata Exchange Format (BMF) [14], Extensible Metadata Platform Format (XMP) [15], DCIM, or even MPEG-7 [16] - the core problem resides in the fact that all these metadata containing precious information are currently not transported as part of the DVB-stream, meaning that there is no possibility to reuse the semantic information of these metadata, mainly used during the production workflow. Television channels certainly could provide this semantic information over an additional interface (e.g., over a Web-based REST-API access), but unfortunately this is currently not the case.

B. Video annotation

Prior to any user interaction with the video stream, a processing mechanism is needed to be able to detect and analyze the actual video content. Here "analysis" describes the process of assigning a unique meaning to a video description and to be able to extract some key features such as who is presenting (name of show host, name of actor), the nature of the program (news, series, cartoon), the topic of the program ("Interview with", "News report", "Music Clip") and also objects or monuments along with their respective names and geo coordinates.

C. Video based analysis

The first straight forward solution is to use video and visual pattern recognition algorithms to do a pixel-based analysis of each video frame as described in [17][18][19] to get the intrinsic context [20][21] of the video (e.g., a plane is landing, or a person is speaking).

Although these approaches might be suitable, they will always need training sets [22] and computational time to consolidate the results by detecting and removing false positives and to, finally, get a fully semantically annotated video frame description [23][24][25]. The prototypical implementation uses the Open CV framework to realize the video based analysis. In order to refine the results, an additional source of information like a MPEG-2 stream is needed.

D. MPEG-2 stream-based analysis

Several types of possible additional sources of information that are embedded in the MPEG-2 stream [26][27][28][29] and used in broadcast systems like DVB were identified. As specified in [29][30], the MPEG-2 stream is delivered over DVB-T and contains several encoded tables and fields enabling contextual information the television receiver is able to decode:

- Electronic Programming Guide (EPG) information stored in the EIT table. Depending on the broadcaster, this information can be very detailed (full description of an episode including the actor's names) or very sparse: only the name of the program along with its schedule is transmitted.
- The channel's Hybrid Broadcast Broadband TV (HbbTV) endpoint URL. Usually a Web site or application URL that can be loaded and displayed by compatible television [31]. This information is extracted from the Application Information Table (AIT) [30].
- Content descriptors that are transmitted usually in form of nibbles which are 4-bit content descriptors that provide a classification of the broadcasted program type (movie, drama, news, sport).
- Teletext and closed captioning information in form of pixel tables (CLUTS) or textual information.

Depending on the country and the broadcaster's allocated bandwidth on a given frequency, the amount of content present in the aforementioned tables might vary, mostly due to the packet sizes in the transmission protocol: broadcasters will logically always privilege the image quality upon transmitting non-video related contents.

The Application Information Table (AIT) contains applications and related information that can be displayed on a

 ISBN: 978-1-61208-353-7

Figure 2. Generation process of a semantic term.

compatible receiver. Within its content descriptor loop, the AIT stores pointers to HbbTV specific information (in some cases also known as Red-Button Service). In most of the cases this pointer is an internet URL that refers to a TV-viewable Web page. By crawling this channel specific Web page additional context can be gained and extracted.

Beside the crawling and extraction of the MPEG-2 tables, another source for our semantic extraction engine is the analysis of Closed Captioning (CC) and subtitles. Subtitles and closed captions were initially introduced for the deaf community to assist them by giving a textual transcription of a scene in form of labels placed over the video. In cases like interviews or documentaries, the closed captioning is a 1:1 transcription of the narrator's spoken text.

All the textual information and extracted context information can be processed by a textual entailment [32] engine that will extract information and deliver semantic concepts and annotations.

E. Mapping of extracted information

Once extracted from the above mentioned streams, the system classifies the extracted terms into several concepts (Person, Object, Monument, etc.), organizes them ontologically (e.g., *[Person[Politician] name: Barack Obama] [isPresidentOf] [Country, name:United States of America]*) and displays them onto the user interface in form of semantic terms. Currently our system will use a classification with following categories: Person (Actor, Politician and Speaker), Object (Car, Building), Companies and fictional Characters. Figure 2 shows how extracted streams are used to generate a visual semantic term defined as *Grabbable*.

F. Audio-based analysis

While the video frame-based analysis is running, an analysis of the audio channel via speech-to-text engine can be used in order to get additional details about the content. The extracted text can then be saved or delivered as a transcript and reused for an information extraction engine. In the case the analysis of the original audio does not deliver enough information, the second possibility is to rely on the Audio Description (AD)

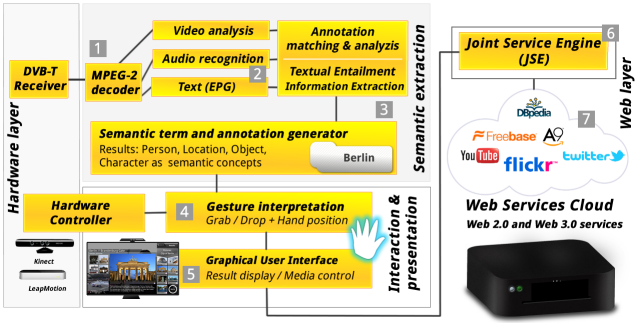

Figure 3. Architecture of the gesture-based semantic TV system.

channel. Along with the original sound of the program, an audio description provides similar to radio drama, a spoken scene description.

III. ARCHITECTURE

The implemented system prototype is based upon a setup-box plugged to a Digital Video Broadcasting Terrestrial (DVB-T) receiver, running a customized UI, and managing interaction hardware like a depth camera (Kinect), a gyration mouse or a finger tracking controller (LeapMotion Controller). The functionality of these components are represented in Figure 3.

The architecture of the prototype system is composed of several abstract processing steps. On the one hand there exists a user-hidden layer of signal analysis and evaluation, shown in the graphic as "Semantic extraction". This layer continuously performs an analysis of the DVB-T signal. As a result semantic terms are generated and can be used as input for a Semantic Web-based information search.

On the other hand, all user-visible processes are initiated by the user on the "Interaction and Presentation" level. This user-centered approach gives the viewer the possibility to access additional information in parallel to the TV program by interacting with the system via a non-disruptive gesture interaction. This gesture allows to trigger a search by simply grabbing a semantic term (e.g., an actor's name) in the system - these terms are called *Grabbables* onto a search field called *Dropzone*. This interaction can be achieved whenever the user wants to get additional information during a TV Program.

Furthermore, the "Web layer" handles the connections to Web-based content. Information from different knowledge domains can be addressed via this interface, as described in detail in Section V.

The following part lists every single processing step and task presented in Figure 3. The role of the complete solution is to:

- Display the DVB-T video signal and decode the information out of the MPEG-2/MPEG-TS stream (1).
- Analyze the MPEG-2 stream and extract information out of the tables to generate corresponding annotations for the broadcasted program (2).
- Create ontologically represented semantic terms and generate graphical equivalents in form of Grabbables (3).

 ISBN: 978-1-61208-353-7

- Interpret gesture interactions and translate them into fully formulated search queries (4).
- Use a graphical overlay principle, to enhance the user's graphical interface with additional Grabbables and multimedia annotated elements e.g., pictures, videos, or shopping items (4-5).
- Connect via Joint Service Engine to Web services, social services like Twitter, and Semantic Web Services such as Freebase and DBpedia (6-7).
- Display search results by using the interaction layer on the graphical user interface (5)

We have chosen this basis for our prototype as we are not restricted in the usage of certain APIs and have full control of both, the UI-side and the stream processing side contrary to closed proprietary solutions proposed by connected TV manufacturers.

IV. User Interface and Interaction

A. Motivation for user interface design

Although aggressively promoted by current TV manufacturers, the TV-app concept is not suitable for a quick search and browsing through the Web even less in the Semantic Web. Moreover if a Web search has to be realized directly from the television set, the painfully and frustrating typing or speaking of a keyword with a remote controller is hindering the interaction. And what happens, if the viewer does not know how to spell or pronounce the name of a building in an interesting reportage about a city? Or the viewer does not know the name of an actor, but can recall that he was starring in an American soap? Only a long search and several switches between TV-apps and the television program might help the curious and interested knowledge hungry viewer. In some cases, this problem can rapidly turn into a decisional problem, as each television broadcaster has its own app with own structures and corporate-designed interfaces leading the user to ask himself which app will be the most suitable for what he is looking for. The interaction problem is even higher when the user is zapping through several channels: must he also switch between different apps and retype his query string each time or change the context of the application manually? Unfortunately, this switching behavior brings a total interaction breach between watching the television program and getting information from the Web.

Starting from these observations, our approach tries to completely redefine the way viewers are interacting with the television by abandoning the current TV-app concept in favor of an intuitive user-centric graphical user interface.

B. User interface

The implemented graphical user interface of the created prototype system is purposely held very easy and follows all along its conception the "10 Feet Design paradigm" [33][34][35] by concentrating the efforts on having a positive trade-off between intuitive user experience, readability and easiness of interaction, so that non-computer specialists will also be able to use the system without having to cope with remote controllers and menus. Figure 4 depicts a screenshot of our

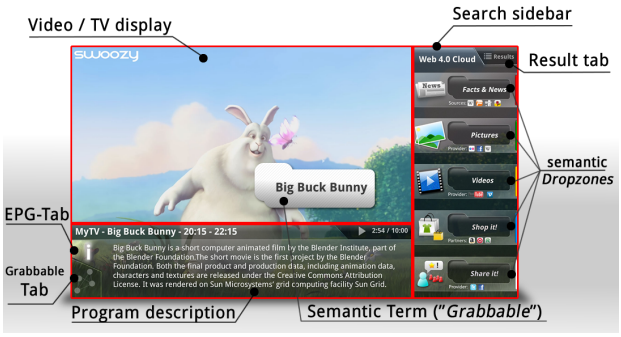

Figure 4. Screenshot of the User Interface.

current semantic television system graphical user interface. The interface consists of a graphical overlay that will be displayed over a video: in the middle of the interface, the regular television program (e.g., received over DVB) or video stream is played. On the right, the user will find a sidebar with five thematic slots (Facts & News, Pictures, Videos, Shop, Share) that internally corresponds to specific service queries. These slots are called *"Dropzones"* and they are able to receive the created semantic terms (*"Grabbables"*). Each displayed *Grabbable* can be grabbed and dropped by the user via gesture interaction. The metaphor of the *Dropzones* is an adaption of the *Spotlets* (graphical intelligent touchscreen-based search agents) mechanism - developed in a previous Web 3.0 based entertainment system [36][37][38][39].

The *Grabbable* dropped in one of the *Dropzones* is always annotated (Figure 5): this means a fact search about a person will have another internal meaning and output than an object search. When searching for facts about a person the search query is enriched by all extracted and represented information of the semantic description (first name, middle name, last name, gender, profession, etc.) which makes the search process of the connected Joint Service Engine (JSE) - described in V - more effective and precise by using better filter options. For example if the user is looking for detailed information about a building additional properties such as the location, its architecture or inauguration date can be returned as each result has a semantic visual representation. This approach follows the "no presentation without semantic representation" paradigm [40][41][42] in usage in numerous multimodal dialog systems [37]. At the bottom of the graphical user interface, the user can either choose one of the generated Grabbables (Figure 4) or switch to the traditional Electronic Program Guide (EPG) view.

This approach breaks with the philosophy of TV-Apps that every app needs its own services. In this implementation, the attached Joint Service Engine (JSE), is able to integrate different Web services, like Wikipedia, DBpedia, Freebase, Flickr or YouTube, simultaneously and it also delivers an orchestration of combined result structures. This means that the viewer will always get a unified result list, as depicted in Figure 6, where combined personal data, such as zodiac sign or portrait pictures of DBpedia and Flickr, is shown as part of the biography. In Figure 6, detailed facts about the famous football player "David Beckham" are displayed on the right side of the user's interface.

Figure 7 shows the results of a search for pictures that was

 ISBN: 978-1-61208-353-7

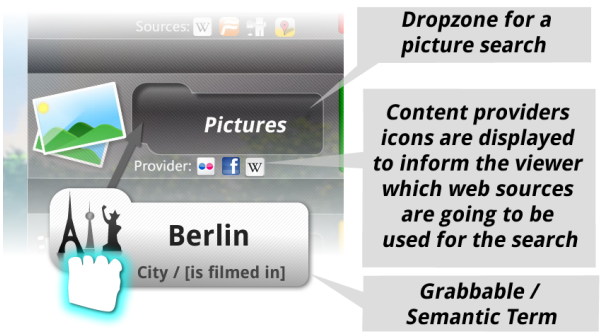

Figure 5. Close-up of a *Dropzone*.

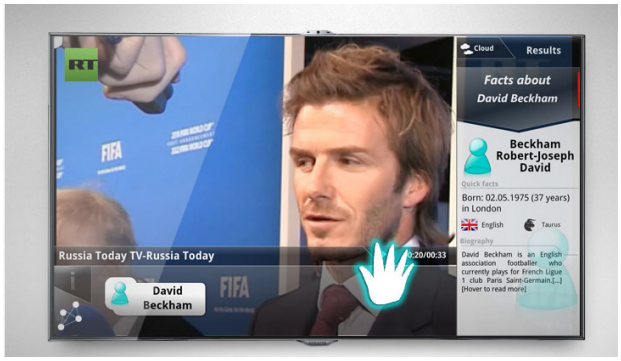

Figure 6. Display David Beckham's biography.

Figure 7. Picture request during a report about Dubai with results coming from different Web sources.

triggered by a location concept named "Dubai". The pictures are retrieved from different databases and extracted by a mashup of Web services (Flickr, Wikipedia and Freebase)

C. Interactions by gestures

Following the same principle of simplicity and easiness of use, we have inbuilt the possibility for the user to interact with the system over gestures: the user only needs to move his hand towards the television screen. At this precise moment, a virtual hand is displayed (Figure 8). The position of the hand can be either tracked over a depth camera like the Microsoft Kinect, or for smaller living rooms by using a finger tracking solution, like the LeapMotion controller device [43].

We have deliberately implemented only two gesture types: *Grab'n'drop* and the *Push*-gesture, as these interactions are

Figure 8. User gesture interaction: a virtual hand allows the user to grab out a semantic term from a video.

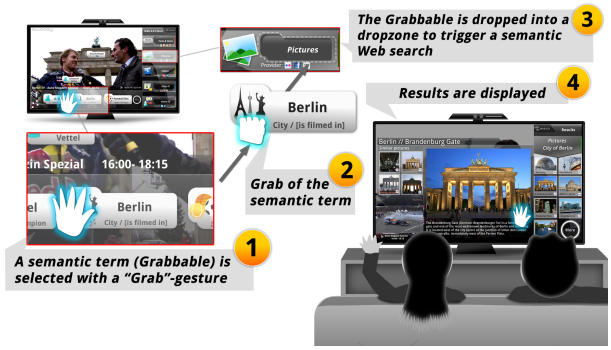

Figure 9. Grab'n'drop interaction steps to start a picture search for the city of Berlin.

simple to realize and do not need a specific user training and do not cause fatigue over time. The *Push* interaction is needed to make a selection and is a simplified metaphor of the traditional mouse click.

Figure 9 describes the interaction workflow. Step #1 shows how the user can grab a semantic term (*Grabbable*) from a sport report featuring Sebastian Vettel during a car show in front of the Brandenburg Gate in Berlin. In our case, the user has selected the term "Berlin" that is internally represented as a location with geo coordinates.

The user now would like to look for pictures of "Berlin". To achieve this, she will take the *Grabbable* (Step #2) and drop it into the Picture *Dropzone* (Step #3); within a few seconds first results coming from the Semantic Web are displayed in form of push-able elements in the right side bar (Step #4).

Beside the easiness of usage of such a system through gesture interaction, the main originality resides also in the fact that without having to type on a keyboard, or to start an additional app, any viewer will be able to rapidly get facts, video or even shopping recommendations during his favorite TV program.

D. Mobile client application

With the mobile application of our approach, depicted in Figure 10 - the mobile Swoozy App (for Android and iOS) -

 ISBN: 978-1-61208-353-7

Figure 10. Swoozy - mobile client application.

multiple users can simultaneously view the same TV program but interact with their own device in parallel. If viewers like to share interesting videos, pictures or facts with the other viewers, they can use the simple "sling-gesture" on their mobile device to transfer these interesting results to the TV with its large display, similarly to the 3D frisbee interaction approach presented by Becker et al. [38], where multimedia content is transferred from mobile devices to a kiosk system.

V. RETRIEVAL OF FACTUAL KNOWLEDGE BASED ON SEMANTIC TECHNOLOGIES

According to the system design, the viewer is supplied with new facts, pictures and videos while watching TV. Therefore, it is absolutely essential to access external sources and to quickly find information that exactly matches to the shown scenery. The presented approach uses a combination of techniques of the Semantic Web to create matching answers, whereas a composition of standard Web services and services of the Semantic Web is serving as knowledge source. However, the heterogeneous aspect of the services and their different Application Programming Interfaces (APIs) represents a challenge for building a correct query and retrieving matching contents. The latter must be adapted in an additional step, so that they can be correctly displayed onto the user's interface.

A. Motivation

As mentioned at the beginning, the video, audio and text analysis extracts knowledge concepts and adds them to predefined ontological structures which can define persons, fictional characters, objects or locations. By the procedure described in this approach and with these prepared input structures, the viewers are able to trigger queries to Web services or Semantic Web Services over simple gesture interaction without the need of special skills, such as programming Web services API or the need of specific database query languages or an RDF(S) query language like SPARQL [44][45]. For non-specialists it would be very hard to formulate such queries. These query languages are primarily used to access the full power of the Semantic Web by allowing a navigation through semantically annotated data sets and enabling the search for instances that corresponds to a given request.

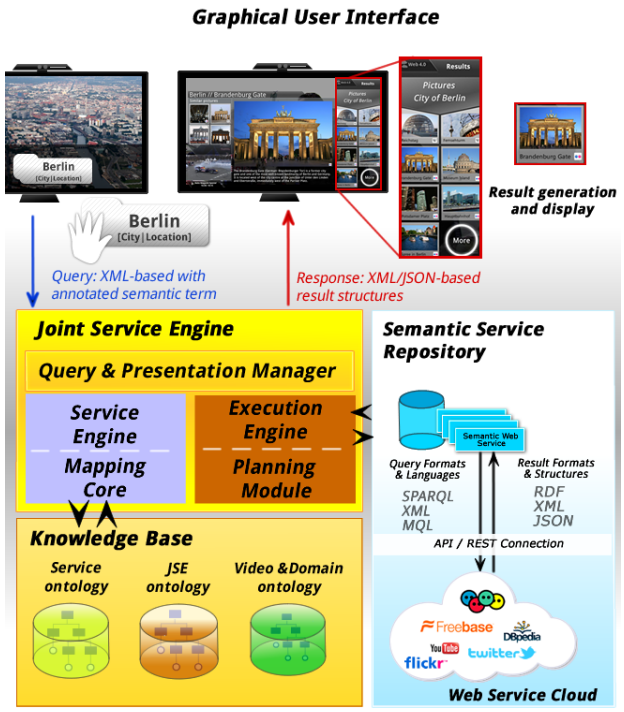

Figure 11. Architecture of the Joint Service Engine.

We assume that the typical viewer does not want to explicitly formulate his search in one of the above-mentioned query languages. That is why the search will be done in the background by using semantically annotated data sets that will be then mapped to the dropped *Grabbable*.

B. Retrieving semantic content

In order to start a search with a *Grabbable*, a dedicated engine was implemented to better solve the tasks of calling heterogeneous services and providing unified semantic results. This engine called *Joint Service Engine* (JSE) is involved in the retrieval of semantic content. The basic idea of the JSE is to use the joint potential of different services to focus information and knowledge. It provides and manages semantic descriptions of various pre-annotated information sources in a local *Semantic Service Repository* that opens up access to sources of different domains. This question answering component internally realizes a judicious orchestration and mashing up of Web 2.0 and Web 3.0 services and provides aggregated results coming from several sources - Web services - as a final result. This result is returned to the client and displayed on the respective user interfaces (television UI and second screen app).

One advantage of this component is that new sources can be added, removed or replaced without hard programmatic dependencies and without stringent dependencies on specific providers of information and their interfaces. Figure 11 shows a specific overview of the architecture design of the JSE.

C. Query processing

The "Query" module of the *Service Engine* [46] retrieves and decomposes the user's query. The produced query structures are formulated according to a terminology defined by domain

 ISBN: 978-1-61208-353-7

```
search topic:
generic concepts = {object (car, building),
                    person (actor, speaker, ..),
                    company,
                    location,
                    fictional character}
query-for:
  similar videos AND/OR pictures
  personal facts AND pictures AND/OR videos
  location AND/OR pictures AND/OR videos
  object facts AND pictures AND/OR videos
  shopping facts AND pictures AND videos
  sharing facts AND pictures AND videos

given properties:
  // depending on concept type
  {first name, middle name, last name, title},
  {gender, profession},
  {characterizing keywords},
  {geo-data (latitude, longitude)},
  {city-name, country-name}
  {building-name}
  {company-facts, company-name, keywords}
```

Figure 12. Query search topics and properties.

ontologies and expressed using a generic template-based query structure as shown in Figure 12. Each individual decomposed query part is mapped to a local meta-representation, the JSE ontology, modeled in OWL [10]. According to the user's query, basic ontological components like individuals are created based on the defined vocabulary of the JSE ontology and the planning module looks for adequate plans that fulfill all of the requested properties. The resolving internal query specifies the *input type* (object, person, fictional character, company, location) specified by *properties* (complete name, keywords, etc.) and *implicit relations* and *search topics* (similar pictures, shopping facts, etc.).

One crucial point in this scenario is the discovery and execution of services. This task is executed by an execution plan which describes the discovery process by specifying which type of services are needed, what kind of domain is addressed, in which order the services have to be executed and all the requirements needed for the matchmaking process occurring in the connected *Semantic Service Repository*. Results of the matchmaking process are ordered lists with adequately ranked information sources. The sequence of individual service calls that must be executed are listed in a scheduling table that needs to be processed by the *Planning Module* and the *Execution Engine*. The *Execution Engine* provides connectors and encapsulates the calls to the REST or API interfaces, by reformulating and using specific query formats like XML or languages, like SPARQL and Metaweb Query Language (MQL). Once all results of different called services are received by the *Execution Engine* an internal mapping process starts a review and reasoning process with the help of additional semantic mapping rules and classifies the results according to the internal JSE domain ontology.

D. Mapping and matching

During the processing chain of the JSE, the content of the described data channels must be repeatedly transformed from one data format to another. The most important step for creating comparable and interoperable data models is the definition of mapping functions between the used concepts. Therefore, the identified data structures must be mapped based on stored mappings that have been defined in a pre-processing phase in a formal description language. For an unambiguous assignment of the models and types of a described element, the mapping functions are specified by categories. The *Mapping Core* achieves these mappings in each component of the JSE. In the "Query" module the user's query input description is mapped to the internal domain ontology that is used for further processing in the planning process. Additionally in the "Execution" module, a mapping of the results of the called external sources to the internal JSE ontology must be fulfilled. Moreover, the spectrum of results of external sources varies from simple XML or JSON structures to complex semantic data structures. In this specific case, formal mapping rules are used to allow a higher quality data type mapping on a more generic level: new instances can be created and linked to each other. Alternatively, a taxonomy of objects can be mapped according to the internal data structure.

E. Service repository

The *Semantic Service Repository* provides access to different types of information sources like Semantic Web Services that cover information stored in external database management systems or Semantic Repositories. In the development of concepts and prototypical implementation [46] detailed service descriptions in OWL-S [47], of freely available sources of knowledge, such as DBpedia, Freebase, Flickr, were integrated. In these Web-based systems information is stored in a structured and manageable form, but can only be accessed by special query languages like SPARQL for DBpedia or MQL in the case of Freebase. The main difference of this approach, compared to conventional database management systems, is the usage of ontologies as a technology to harmonize and store semantically structured data: each concept defines and classifies information and also adds implicit knowledge characterized by its name and position in a hierarchy or taxonomy [46].

The JSE closes the gap between pure RESTful service calls and factual knowledge extracted from Semantic Web Services like Freebase or DBpedia, by mapping results and their respective annotations syntactically and semantically well-defined to a domain ontology.

F. Output presentation

The last step of the processing is done by the *Presentation Manager* which will encapsulate and transform the semantic annotations in a standardized result structure. The contents of the delivered result structures are displayed on the graphical user interface (television screen) after a parse process. Depending on the user's query, e.g., a media search, other different structured output formats (RDF, XML, JSON, etc.) can be served by the *Presentation Manager* module. This module uses filter rules and generic declarative element-based mapping techniques to create the resulting structures from the internal domain ontology and returns these structures to connected client platforms. This procedure allows a parallel distributed output: both second screen and television systems are fed with the results coming from the *Presentation Manager*. With this parallel output processing a cross-media interaction is possible.

 ISBN: 978-1-61208-353-7

VI. Conclusion and future work

With *Swoozy*, the prototypical implementation of this approach, we demonstrated that, through a seamless combination of gesture-based interaction, video information coming directly from the broadcaster and the Joint Service Engine, it is possible to provide a novel way to interact with video contents. Through this approach, television enters into a new dimension in which viewers will receive additional information and knowledge about the persons, locations, objects featured in a video or a television program.

The *Swoozy* concept is not only applicable for the sole field of television, but can also be used for other video-based systems such as interactive e-Learning systems, video casts or even online university courses, where the semantic terms would be mathematical formulas or technical concepts.

We believe that the concept of semantic television will turn television into an appealing and ludic knowledge provider and will give a brand new dimension to interactive connected television systems in the future. Moreover in addition to the input modalities (Microsoft Kinect and LeapMotion controller) used in *Swoozy*, we consider extending our gesture-based approach to SmartWatches.

References

[1] SevenOneMedia, "HbbTV macht TV clickbar," 2013.

[2] L. Aroyo, L. Nixon, and L. Miller, "NoTube: the television experience enhanced by online social and semantic data," in Consumer Electronics-Berlin (ICCE-Berlin), 2011 IEEE International Conference. IEEE, 2011, pp. 269–273.

[3] Y. B. Fernandez, J. J. Pazos Arias, M. L. Nores, A. G. Solla, and M. R. Cabrer, "AVATAR: an improved solution for personalized TV based on semantic inference," Consumer Electronics, IEEE Transactions on, vol. 52, no. 1, 2006, pp. 223–231.

[4] J. Kim and S. Kang, "An ontology-based personalized target advertisement system on interactive TV," Multimedia Tools and Applications, vol. 64, no. 3, 2013, pp. 517–534.

[5] B. Makni, S. Dietze, and J. Domingue, "Towards semantic TV services a hybrid semantic web services approach," 2010, [Retrieved: July 2014].

[6] T. Berners-Lee, J. Hendler, and O. Lassila, "The Semantic Web," Scientific American, 2001, [retrieved: July 2014]. [Online]. Available: http://www.jeckle.de/files/tblSW.pdf

[7] G. Klyne and J. J. Carroll, "Resource Description Framework (RDF): Concepts and Abstract Syntax," W3C Recommendation, 2004. [Online]. Available: http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/

[8] P. Hitzler, M. Krötzsch, S. Rudolph, and Y. Sure, Semantic Web: Grundlagen. Springer Berlin Heidelberg, 2008.

[9] R. Cyganiak, D. Wood, and M. Lanthaler, "RDF 1.1 Concepts and Abstract Syntax," W3C Recommendation, 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/

[10] P. F. Patel-Schneider, P. Hayes, and I. Horrocks, "OWL Web Ontology Language Semantics and Abstract Syntax," Feb. 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/2004/REC-owl-semantics-20040210/

[11] M. C. Surez-Figueroa, G. A. Atemezing, and O. Corcho, "The landscape of multimedia ontologies in the last decade," Multimedia Tools and Applications, vol. 62, no. 2, 2013, pp. 377–399.

[12] M. Lux, W. Klieber, and M. Granitzer, "Caliph & Emir: semantics in multimedia retrieval and annotation," in Proceedings of the 19th International CODATA Conference. Citeseer, 2004, pp. 64–75.

[13] M. Lux and M. Granitzer, "Retrieval of MPEG-7 based semantic descriptions," in In Proceedings of BTW-Workshop WebDB Meets IR, 2004.

[14] Institut für Rundfunktechnik, "Broadcast Metadata Exchange Format," BMF 2.0, 2012, [retrieved: July 2014]. [Online]. Available: http://bmf.irt.de/

[15] Adobe, "XMP - Adding intelligence to media," 2012, [retrieved: July 2014]. [Online]. Available: http://www.adobe.com/devnet/xmp.html

[16] J. Martnez, R. Koenen, and F. Pereira, "MPEG-7: the generic multimedia content description standard - part 1," Multimedia, IEEE, vol. 9, no. 2, 2002, pp. 78–87.

[17] S. Bloehdorn et al., "Semantic Annotation of Images and Videos for Multimedia Analysis," in The Semantic Web: Research and Applications, ser. Lecture Notes in Computer Science, A. Gómez-Pérez and J. Euzenat, Eds. Springer Berlin Heidelberg, 2005, vol. 3532, pp. 592–607.

[18] E. Sgarbi and D. L. Borges, "Structure in soccer videos: detecting and classifying highlights for automatic summarization," in Progress in Pattern Recognition, Image Analysis and Applications. Springer, 2005, pp. 691–700.

[19] W. Shao, G. Naghdy, and S. Phung, "Automatic image annotation for semantic image retrieval," in Advances in Visual Information Systems, ser. Lecture Notes in Computer Science, G. Qiu, C. Leung, X. Xue, and R. Laurini, Eds. Springer Berlin Heidelberg, 2007, vol. 4781, pp. 369–378.

[20] L. Ballan, M. Bertini, and G. Serra, "Video Annotation and Retrieval Using Ontologies and Rule Learning," IEEE MultiMedia, vol. 17, no. 4, 2010, pp. 80–88.

[21] U. Arslan, M. E. Dönderler, E. Saykol, O. Ulusoy, and U. Güdükbay, "A Semi-Automatic Semantic Annotation Tool for Video Databases," in Proc. of the Workshop on Multimedia Semantics (SOFSEM 2002). Milovy, Czech Republic, ser. SOFSEM-2002, 2002, pp. 1–10.

[22] G. Quénot, "TRECVID 2013 Semantic Indexing Task," 2013.

[23] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," Pattern Analysis and Machine Intelligence, IEEE Transactions on, vol. 32, no. 9, 2010, pp. 1627–1645.

[24] C. Snoek, D. Fontijne, Z. Z. Li, K. van de Sande, and A. Smeulders, "Deep Nets for Detecting, Combining, and Localizing Concepts in Video," 2013.

[25] L. J. Li, H. Su, L. Fei-Fei, and E. P. Xing, "Object bank: A high-level image representation for scene classification & semantic feature sparsification," in Advances in neural information processing systems, 2010, pp. 1378–1386.

[26] T. Dajda, M. Cislak, G. Heldak, and P. Pacyna, "Design and implementation of the electronic programme guide for the MPEG-2 based DVB system," 1996.

[27] C. Peng and P. Vuorimaa, "Decoding of DVB Digital Television Subtitles," Applied Informatics Proceedings - No.3, 2002, pp. 143–148.

[28] M. Dowman, V. Tablan, H. Cunningham, C. Ursu, and B. Popov, "Semantically enhanced television news through web and video integration," in Second European Semantic Web Conference (ESWC2005). Citeseer, 2005.

[29] "Digital Video Broadcasting (DVB) Subtitling systems," European Standard ETSI EN 300 743, European Broadcasting Union, 2014.

[30] "Specification for Service Information (SI) in DVB systems," European Standard ETSI EN 300 468, European Broadcasting Union, 2014.

[31] K. Merkel, "HbbTV - Status und Ausblick," 2012, [retrieved: July 2014]. [Online]. Available: http://www.irt.de/webarchiv/showdoc.php?z=NTgwNyMxMDA1MjE1I3BkZg==

[32] R. Wang and G. Neumann, "Recognizing textual entailment using sentence similarity based on dependency tree skeletons," in Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, ser. RTE '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 36–41.

[33] R. Cardran, K. Wojogbe, and B. Kralyevich, "The Digital Home: Designing for the Ten-Foot User Interface," 2006, [retrieved: July 2014]. [Online]. Available: http://channel9.msdn.com/Events/MIX/MIX06/BTB029

[34] Samsung, "Design Principles for Creating Samsung Apps Content," 2013, [retrieved: July 2014]. [Online]. Available: http://www.samsungdforum.com/UxGuide/2013/01_design_principles_for_creating_samsung_apps_content.html#ux-01

 ISBN: 978-1-61208-353-7

[35] D. Loi, "Changing the TV industry through user experience design," in Design, User Experience, and Usability. Theory, Methods, Tools and Practice, ser. Lecture Notes in Computer Science, A. Marcus, Ed. Springer Berlin Heidelberg, 2011, vol. 6769, pp. 465–474.

[36] D. Porta, M. Deru, S. Bergweiler, G. Herzog, and P. Poller, "Building multimodal dialogue user interfaces in the context of the internet of services," in Towards the Internet of Services: The Theseus Program, ser. Cognitive Technologies, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, and T. Widenka, Eds. Springer Berlin Heidelberg, 2014, pp. 149–168.

[37] D. Sonntag, M. Deru, and S. Bergweiler, "Design and implementation of combined mobile and touchscreen-based multimodal web 3.0 interfaces," in Proceedings of the International Conference on Artificial Intelligence, ser. ICAI-09, July 2009, pp. 974–979.

[38] T. Becker et al., "A unified approach for semantic-based multimodal interaction," in Towards the Internet of Services: The Theseus Program, ser. Cognitive Technologies, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, and T. Widenka, Eds. Springer Berlin Heidelberg, 2014, pp. 135–148.

[39] S. Bergweiler, M. Deru, and D. Porta, "Integrating a Multitouch Kiosk System with Mobile Devices and Multimodal Interaction," in Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces, ser. ITS-2010, ACM. 1515 Broadway New York, New York 10036: ACM, 2010.

[40] W. Wahlster and A. Kobsa, "User models in dialog systems," in User Models in Dialog Systems, ser. Symbolic Computation, A. Kobsa and W. Wahlster, Eds. Springer Berlin Heidelberg, 1989, pp. 4–34.

[41] A. Kobsa, "Generic User Modeling Systems," User Modeling and User-Adapted Interaction, vol. 11, no. 1-2, 2001, pp. 49–63.

[42] N. Reithinger et al., "A look under the hood: design and development of the first SmartWeb system demonstrator," in Proceedings of the 7th international conference on Multimodal interfaces. ACM, 2005, pp. 159–166.

[43] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the Accuracy and Robustness of the Leap Motion Controller," Sensors, vol. 13, no. 5, 2013, pp. 6380–6393.

[44] "SPARQL query language for RDF," W3C Recommendation, 2008, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[45] "SPARQL 1.1 query language," W3C Recommendation, 2013, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[46] S. Bergweiler, "Interactive service composition and query," in Towards the Internet of Services: The Theseus Program, ser. Cognitive Technologies, W. Wahlster, H.-J. Grallert, S. Wess, H. Friedrich, and T. Widenka, Eds. Springer Berlin Heidelberg, 2014, p. 480.

[47] D. Martin et al., "OWL-S: Semantic Markup for Web Services," W3C Submission, 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/

 ISBN: 978-1-61208-353-7