Foundations of Semantic Television

Design of a Distributed and Gesture-Based Television System

Simon Bergweiler and Matthieu Deru

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

Abstract—**The innovations in information and communication technologies change our daily life and the way how to interact with intelligent systems. Powerful computers are becoming smaller and are integrated almost anywhere, even in televisions. Today's connected television systems are offering a lot of technical functionalities including these, which are currently integrated in smartphones. In this article, we describe an innovative approach in form of an intelligent television system named *Swoozy*, which enables viewers to discover extended information, such as facts, images, shopping recommendations or video clips about the currently broadcast TV program by using the power of technologies of the Internet and the Semantic Web. 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. These questions are very often 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.**

Index Terms—interactive television system; Semantic Web Technologies; video annotation; gesture-based interaction.

I. Introduction

With the growing popularity of smartphone applications (apps) a new trend slowly appeared to integrate these capabilities into television systems. In fact, the so-called connected television systems provide a wide range of technical capabilities that opens the viewers new possibilities to communicate and interact with the Internet and its services with similar features their smartphones would currently provide. This article describes an innovative approach in form of an intelligent television system named *Swoozy* [1]. This self-designed and implemented system enables viewers to discover extended information, such as facts, images, shopping recommendations or video clips about the currently broadcast TV program by using the power of technologies of the Internet and the Semantic Web.

A study conducted by the German marketer for audiovisual media SevenOneMedia [2] 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 activity 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: users must install a lot of single apps, for example, one for searching videos another one for images in order to get the information they are looking for. Another technology widely spread in Europe is the Hybrid Broadcast Broadband TV standard (HbbTV) that certainly offers viewers an alternative to apps, but is currently still limited in interaction and search possibilities. These trends and technologies are described in detail in Section III.

The usage of these solutions also reveals another problem: the constant switches 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.

To solve these interaction issues, the discussed approach presents a new way how viewers can interact with additional content while watching a TV program. In fact, with our solution, they are able to search in parallel for information in the Web and easily browse through the found results without an interaction breach. In its first version, 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, to get precise additional information in relation to the current shown scenery, like further videos, text or news articles, pictures, and furthermore shopping recommendations.

Whereas system prototypes like NoTube [3] and others [4][5][6] are using the Semantic Web for detecting possible matches between the watched program and other Web-based contents to only offer a personalized TV access, our approach uses semantic technologies on several levels. The first level is the extraction of knowledge and concepts from an ordinary non pre-annotated Digital Video Broadcasting (DVB) data stream (also called video signal). From this DVB data stream, the required information is extracted and transferred via matching rules into annotations. Over an intuitive dedicated gesture-based graphical TV interface, presented in Section V, the viewer can easily trigger a search using semantic queries. These queries are finally processed by a specially designed and implemented backend 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 VI.

The prototype was implemented in form of set top box-based software solution to demonstrate the technical 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 the TV signal. A technical overview of currently available TV systems including the functioning of HbbTV is given in Section III. This state of the art is necessary to better delimit the core aspect of our approach from the ones, which are commercially available. Section IV presents in detail each implemented module used during the extraction process. In Section V, the choices for the design of the user interface are motivated and the method how gesture interactions lead to a semantic search is presented. Section VI 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 [7] and its related 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 [8], and common Web service interfaces in form of service mashups [9].

The World Wide Web Consortium (W3C) has declared ontologies as an open standard for describing information of an application domain and also defined appropriate ontological description languages such as RDF(S) [8][10] and OWL [11]. Ontologies, as specification languages have been specially developed for a usage within the Semantic Web and mainly consists of concepts and relations. Relations organize concepts hierarchically and put them together in 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 more precisely a mapping

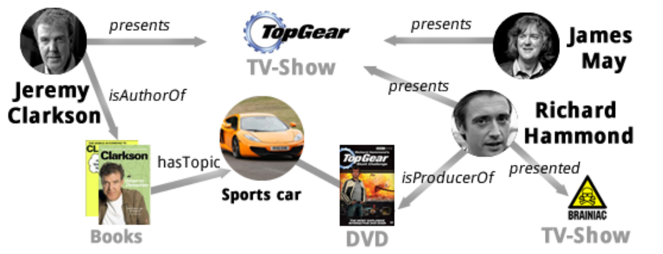


Fig. 1: Discovering new semantic relations in a TV domain.

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 can be classified [12]. This gain of knowledge out of a video can only be realized by video-based annotations: in our system we call these semantic terms or *Grabbables*.

Although several tools [13][14] and solutions exist for embedding metadata and annotations along with video - most of them are working with XML-based annotation formats like Broadcast Metadata Exchange Format (BMF) [15], Extensible Metadata Platform Format (XMP) [16], DCIM, or even MPEG-7 [17] - 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, as these are mainly used during the production workflow and not made available for further usage. 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. Semantic Web services

Semantic Web services play a central role in the presented approach as they will deliver additional contents. To achieve a correct and coherent mapping between the semantic terms and what has to be found (e.g., biography, pictures), an internal Web service ontology is needed. The latter will define how the Web services have to be accessed in term of interfaces and result types. One language to internally describe these semantic Web Services is the Web service ontology language (OWL-S) [18]. OWL-S is based on the Web Ontology Language (OWL) [19], a recommendation of the W3C, and extends its set to structures that include properties, specificity and dependencies of the Web service and express them in machine-readable and processable structures.

A concrete service description in OWL-S is divided into three parts: the *service profile*, a *service model*, and *service grounding*. Primarily the service profile is used for service discovery and describes what the service does. It contains information about the organization that provides the service, the preconditions, input and output values, and effects, as well as the features and benefits of the service. Once a service has been selected, the *service profile* is no longer used. For the concrete process of service execution the description defined in the *service model* is used. Figure 2 shows the main concepts and relations of a service description defined in OWL-S.

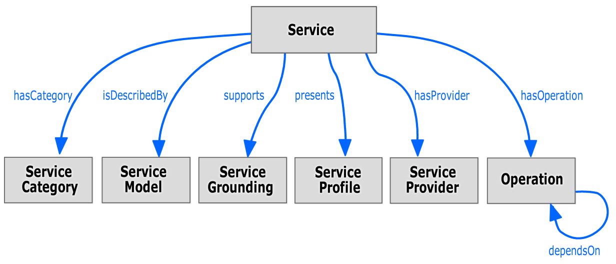


Fig. 2: Main concepts of the Web service ontology language OWL-S.

The *service model* describes the actual execution process of a service. Here, this process description consists of simple atomic processes or complex composite processes that are sometimes abstract and not executable. The process describes the individual use of the service by clients by specifying input and output data, preconditions and effects.

The *service grounding* provides detailed technical communication information on protocols and formats as well as addressing details. Furthermore, the grounding model provides a direct link or mapping between the service model and the technical service execution level. For example, implementation details like input and output messages of the *service model* are translated into corresponding elements of the service description language. The W3C recommends and specifically describes in its member submissions WSDL, but other groundings are also possible. For a better understanding of this recommendation, it is important to know that W3C member submissions serve as input to the standards process. These descriptions contain concrete information for the service implementation and realization by enabling a direct link between the grounding and the WSDL elements. In their research articles, Sirin et al. describe their prototypical implementation to directly combine OWL-S with actual executable invocations of WSDL [20][21].

The in here presented approach uses the JSE and its *Semantic Service Repository* module, described in detail in Section VI-F, that is based on the preliminary work in the area of semantic Web services modeling with grounding in WSDL and expands the approach to lightweight REST-based interfaces with their service descriptions in WADL [22][23].

C. 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 geographical coordinates.

D. 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 [24][25][26] to get the intrinsic context [27][28] 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 [29] and computational time to consolidate the results by detecting and removing false positives and to, finally, get a fully semantically annotated video frame description [30][31][32]. The prototypical implementation of *Swoozy* uses the Open CV framework to realize the video-based analysis. In order to refine the results, an additional source of information like a DVB MPEG-2 stream is needed.

E. MPEG-2 stream-based analysis

Several types of possible additional sources of information that are embedded in the MPEG-2 stream [33][34][35][36] and used in broadcast systems like DVB were identified. As specified in [36][37], 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 [38]. This information is extracted from the Application Information Table (AIT) [37]. The functioning of HbbTV is described in detail in Section III-B.
- 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 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 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 textual entailment [39] and Named Entity Extraction engines that will extract information and deliver semantic concepts as annotations.

F. *Named Entity Recognition*

Named Entity Recognition consists in extracting information out of an unstructured text. In our case, as we want to extract detailed information about a currently running TV program, it is necessary to extract it from EPG or program description. To achieve this, we rely on the Java-based implementation of the Stanford Named Entity Recognition [40] that is able to extract and label out of a text, 7 types (also named classes or concepts): Time, Location, Organization, Person, Money, Percent and Date. The result of this analysis can be delivered in XML form where each tag includes the detected concept.

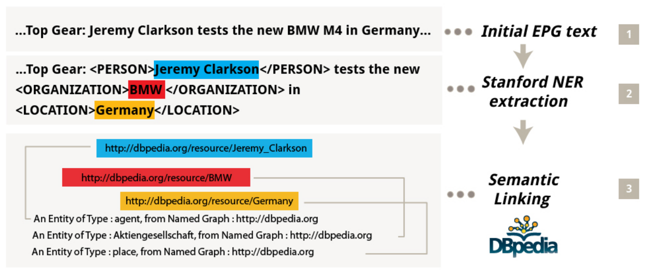

...Top Gear: Jeremy Clarkson tests the new BMW M4 in Germany...
Initial EPG text
1
...Top Gear: <PERSON>Jeremy Clarkson</PERSON> tests the new <ORGANIZATION>BMW</ORGANIZATION> in <LOCATION>Germany</LOCATION>
Stanford NER extraction
2
http://dbpedia.org/resource/Jeremy_Clarkson
http://dbpedia.org/resource/BMW
http://dbpedia.org/resource/Germany
An Entity of Type : agent, from Named Graph : http://dbpedia.org
An Entity of Type : Aktiengesellschaft, from Named Graph : http://dbpedia.org
An Entity of Type : place, from Named Graph : http://dbpedia.org
Semantic Linking
3
DBpedia

Fig. 3: Steps executed during the semantic annotation process.

Based on this first-pass analysis concepts can be linked to DBPedia [41] entities thus leading to a fully semantically annotated structure as depicted in Figure 3. This structure can then be used to define the type of *Grabbable* - a visual semantic term - to be displayed.

G. *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 4 shows how extracted streams are used to generate a visual semantic term defined as *Grabbable*.

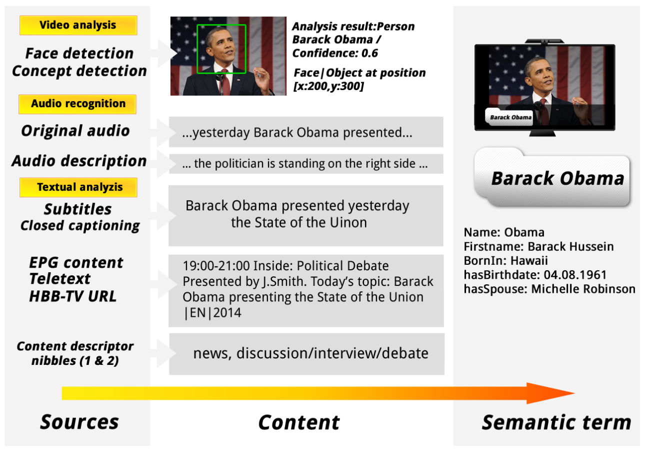

Video analysis
Face detection
Concept detection
Analysis result:Person
Barack Obama /
Confidence: 0.6
Face|Object at position
[x:200,y:300]
Audio recognition
Original audio
Audio description
...yesterday Barack Obama presented...
... the politician is standing on the right side ...
Textual analysis
Subtitles
Closed captioning
Barack Obama presented yesterday the State of the Uinon
EPG content
Teletext
HBB-TV URL
19:00-21:00 Inside: Political Debate
Presented by J.Smith. Today's topic: Barack Obama presenting the State of the Union
|EN|2014
Content descriptor nibbles (1 & 2)
news, discussion/interview/debate
Barack Obama
Barack Obama
Name: Obama
Firstname: Barack Hussein
BornIn: Hawaii
hasBirthdate: 04.08.1961
hasSpouse: Michelle Robinson
Sources
Content
Semantic term

Fig. 4: Generation process of a semantic term.

H. *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 that the analysis of the original audio does not deliver enough information, the second possibility is to rely on the Audio Description (AD) channel. Along with the original sound of the program, an audio description provides similar to radio drama, a spoken scene description.

III. Interactive Television Systems Today: A Compact Overview

Combining television with the Internet is not a completely new idea. First systems in the late 1990's, like WebTV, later MSN TV, have shown that there is a real added value offering a unified access to email and to the Internet over a single box connected to a television device. These systems were also announcing the first wave of interaction television systems so called SmartTVs or Connected TVs that would let viewers surf and access specific services while watching their favorite TV programs. Starting from the early 2010's the market of connected TVs started to be more active and appealing through the fact that several TV manufacturers are integrating new platform technologies (e.g., Android TV, WebOS, Linux), and new interaction paradigms but also allowing third-party developers to implement their own TV-based apps.

The following section presents an overview of the state of the art in the field of interactive television systems including app-based systems. In the last part of this section, the HbbTV technology will be presented as part of an independent and broadcaster initiated approach. This overview will also focus on design, interaction and technical implementation aspects of those platforms to better understand how the *Swoozy* approach radically differs from current commercially available systems.

A. App-based Smart TV

App-based connected TV systems are systems that are mainly focused on delivering additional content in parallel with the live television signal in form of applications or apps. This approach is borrowed from the mobile phone field, commonly known as smartphones, in which apps are very popular and playing the central interaction role between users and connected information sources. Numerous manufacturers are supporting the implementation and distribution of apps over their platforms. The following four major platforms are currently playing an essential role in the connected TV microcosm:

1) *WebOS (LG)*
WebOS is a relatively new system (2014) in the field of connected TV, but is now a well established successor of LG's first Smart TV platform called NetCast. Originally built for the now defunct Palm Pre, WebOS was licensed and brought on television by LG. Main feature of this platform is to enable users to install apps and control them over the Magic Remote - a gyroscope-based remote controller - via HTML5-based applications. The available apps are visually placed in form of icons into a bar that is present at the bottom of the television's video area. From a developer point of view, the WebOS platform allows to use third-party application programming interfaces (APIs), Javascript frameworks such as EnyoJS to build up a generic user interface and common HTML5 tools. The Software Development Kit (SDK) delivered in form of an Eclipse-based Integrated Development Environment (IDE) and an emulator helps to develop WebOS apps without having to physically install these onto the television. LG specific APIs like the Luna API are allowing a restricted access to hardware and system specific information, but unfortunately these APIs do not provide any access to EPG, channel information needed to build up a live-context centric app.

2) *Samsung TVs*
Samsung belongs to the first major manufacturers which have pushed the Smart TV concept onto the mass consumer market. Users can either use their remote controllers or hands to control a virtual cursor and to interact within apps. While using for a few years the same platform, Samsung began in the late 2014's to switch their hardware components to Tizen. The latter is an open-source operating system for numerous devices like mobile phones, wearables and TVs. This leads to the situation that currently (May 2015) two distinct SDKs are provided by Samsung: the Samsung TV SDK and the Samsung Tizen SDK. For third-party application development an Eclipse IDE framework is provided as well as an emulator to better help developers in testing their application. Over the Javascript-based Tizen Web Device API it is possible to access to additional TV channel information such as EPG or currently running show names. To support a unified visual interaction concept, a Javascript UI-Framework called *Caph* offers the possibility to easily and quickly develop UIs and apps for Tizen-based Samsung TV along with an Eclipse-based IDE for using HTML5-based UI components.

3) *Android-based TV systems*
After the early marketing difficulties of Google TV, Google has decided to persist in the television field by releasing a revamped Android-based television system. Android TV systems are currently either present in form of set-top boxes like the Nexus Player or integrated into television hardware systems sold by Philips (TP Vision). Another Android-based set-top box system is the Amazon Fire TV. All built upon Android, these systems offer viewers the possibility to interact with all apps present in the Google Play or Amazon App Store via either a remote controller, a mobile phone or even a smartwatch. The tight integration of Google-based services offers also additional features, like the online-speech recognition through the remote controller. According to the Android TV APIs it is possible to access to channel specific information like EPG although there are currently no devices on the market supporting these features. Applications for Android TV or Amazon Fire TV can be implemented in Java by using the official Android Studio IDE and the specific SDKs although other programming languages like HTML5 are also supported by the platform.

4) *SmartTV Alliance*
The SmartTV Alliance was founded in 2012 by LG and TP Vision and had as goal to setup standards and specifications for Smart TVs. Meanwhile the alliance counts over fifteen members including TV manufacturers like LG, Philips, Toshiba, Panasonic, Vestel or even IBM, which are providing white books and specifications about Smart TV Apps development. The SmartTV Alliance currently suggests to use HTML5 technologies for the implementation and to achieve this, it also provides an unified SDK for developers under the commercial motto "Build once, publish everywhere". In fact, another task of the SmartTV Alliance is to provide tooling to quickly develop apps that will then run on all the partner's hardware. The SDK includes an emulator and an Eclipse IDE for creating TV apps. A review process is then necessary before the application is accepted and dispatched to all the Smart TV alliance's devices. A full specification [42] describes each element developers should take care of during the development of HTML5-based TV apps.

5) *Initial Summary - App-based Smart TV*
Although all the platforms are providing tools and APIs

to implement apps none of them are really providing a seamless integration of both apps and live television video: in fact there is always an interaction breach between the transition from television mode to the app mode. Moreover, during our tests overlaying the current live-video with additional content was not possible. The access to live-video information like EPG or controlling the channels is very limited, on some platforms even impossible. All these elements led us to check whether HbbTV could help the viewer to access this missing information.

B. *HbbTV*

Hybrid Broadcast Broadband TV (HbbTV) is since 2010 a standard that specifies and defines interactive applications and additional interactive content that can be displayed by a hybrid television system [43]. It is often considered as an interactive and more appealing version of the Teletext. HbbTV was founded by the HbbTV consortium comprising several European television broadcasters (e.g., TF1, Canal+, France Television Group, ARD, ZDF). HbbTV is widely used in Europe and currently in tests in other countries like Australia or China. HbbTV also defines standardized interfaces for the usage of Internet-based technologies like IP-TV, video streaming and interactive Web-based value-added services and their technical integration into upcoming television hardware. An example of HbbTV services and the user interface is depicted in Figure 5.

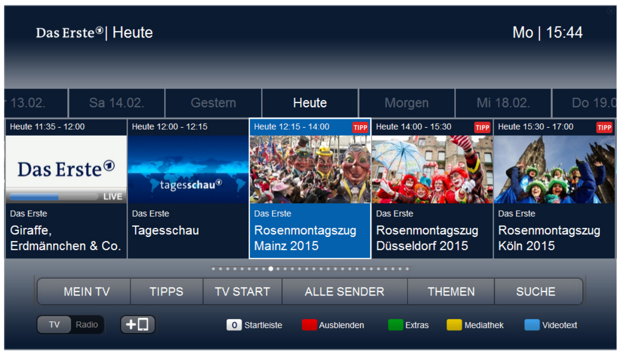


Fig. 5: Example of HbbTV services (source: ARD).

Delivering technical specifications on how interactive services should be broadcasted and offered to viewers is also part of the HbbTV activities.

Hybrid means, HbbTV works in two modes. The *Broadcast* mode is controlled by the television broadcaster: he is in charge of packing into the video stream additional interactive applications. When the receiver decodes the stream, a message invites the viewer to press a button on their remote control. This is known as *Red Button* application in reference to the BBC Red Button service that since 1999 offers viewers the possibility to access additional programs and information over their television hardware. Once this button is pressed, the application is loaded (either from the video stream or from the Internet) and appears on the viewer's display.

In order to get live up-to-date information, HbbTV can rely on its second mode - *Broadband* - which will provide the link to the Internet and to the broadcaster's own online information sources (database or applications) as depicted in Figure 6.

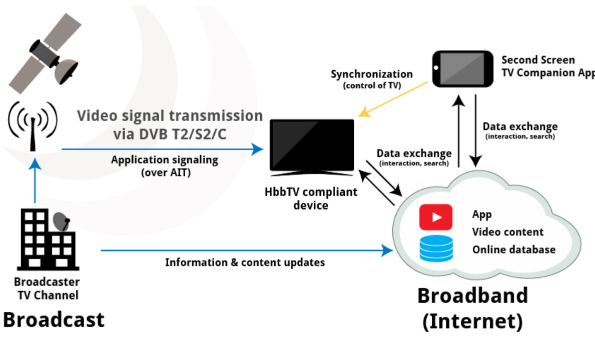

Video signal transmission via DVB T2/S2/C
Application signaling (over AIT)
Synchronization (control of TV)
Second Screen TV Companion App
Data exchange (interaction, search)
Data exchange (interaction, search)
HbbTV compliant device
App
Video content
Online database
Information & content updates
Broadcaster TV Channel
Broadcast
Broadband (Internet)

Fig. 6: HbbTV2 Architecture and technical overview.

In its second specification version, HbbTV 2 that is due in 2015, the integration of second screen and screen companions plays a major role for the broadband mode. In fact, the Internet connection is used to synchronize and push content onto the television, thus allowing a synchronization between the TV program, the information offered over the TV-based HbbTV services and the content displayed on the viewer's tablet. Although being built upon HTML-5 technologies, HbbTV contents and applications are completely working in their own ecosystem. It is not possible for third-party developers to integrate their own app or services, as only the broadcaster can decide when and which kind of services will be provided to viewers at a given moment. Moreover, from a user interaction perspective, HbbTV contents can have visually very different designs, as each channel will adopt a different layout or colors for their channel-specific applications. This leads from a user experience perspective to a lack of unified interaction and to confusion: the viewer must adapt to a new UI (including new functions and services) each time he will call the HbbTV application of a different channel.

C. *Summary*

Although some connected TVs rely on standard Web technologies like HTML5, the manufacturer's restrictions concerning APIs and technical aspect, do not allow third-party developers to currently leverage the full possibilities of interactive television systems. Moreover, TV-based apps running on connected TVs and the HbbTV services do not offer a sufficient and satisfiable approach to leverage the full possibilities of semantic web. Only a limited set of interaction with the content is offered to viewers, and this, within a very closed and predetermined field.

Starting from these facts and observations, the decision was made to implement a TV system that would really enable viewers to intuitively access all internet resources over user-centered interactions without any technical limitations.

IV. Architecture

The implemented system prototype is based upon a set top box plugged to a Digital Video Broadcasting Terrestrial (DVB-

T) receiver, running a customized UI, and managing interaction hardware components like a depth camera (Microsoft Kinect), a gyration mouse or a finger tracking controller (LeapMotion Controller). The functionality of these components are represented in Figure 7.

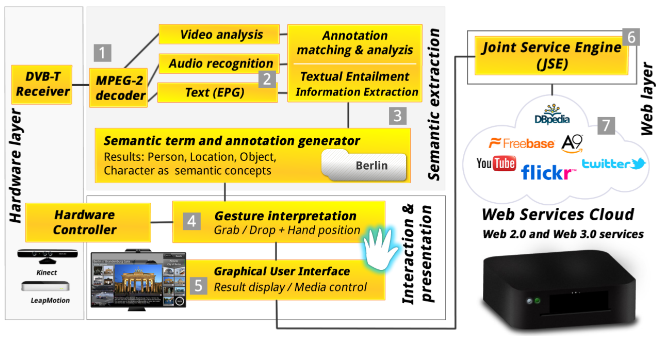

Hardware layer
DVB-T Receiver
1
MPEG-2 decoder
Video analysis
Audio recognition
2
Text (EPG)
Annotation matching & analyzis
Textual Entailment Information Extraction
3
Semantic extraction
Semantic term and annotation generator
Results: Person, Location, Object, Character as semantic concepts
Berlin
Hardware Controller
Kinect
LeapMotion
4
Gesture interpretation
Grab / Drop + Hand position
5
Graphical User Interface
Result display / Media control
Interaction & presentation
Joint Service Engine (JSE)
6
Web layer
DBpedia
7
Freebase
A9
YouTube
flickr
twitter
Web Services Cloud
Web 2.0 and Web 3.0 services

Fig. 7: Architecture of the gesture-based semantic TV system.

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" layer. 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 context of our system's approach, these terms are called *Grabbables* and must be dragged-and-dropped onto a dedicated search field element on the user interface, called *Dropzone*. This drag-and-drop 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 VI.

The following part lists every single processing step and task, identified by its number, presented in Figure 7. 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).
- Interpret gesture interactions and map 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 JSE to Web services, social services like Twitter, and Semantic Web Services such as Freebase or 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.

V. User Interface and Interaction

A. Motivation for user interface design

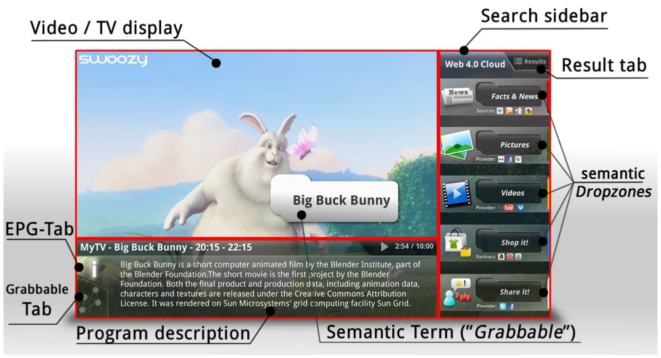
Although aggressively promoted by current TV manufacturers, we believe that the TV app concept is not suitable for a quick search and browsing through the Web even less in the Semantic Web as described previously in Section III. Moreover, if a Web search has to be realized directly from the television set, the painfully and frustrating typing or even 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 report 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 decision 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" [44][45][46] 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 8 depicts a screenshot of our 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 Semantic Web based entertainment system [47][48][49][50].



MyTV - Big Buck Bunny - 20:15 - 22:15 2:54 / 10:00

Big Buck Bunny is a short computer animated film by the Blender Institute, part of the Blender Foundation.The short movie is the first project by the Blender Foundation. Both the final product and production data, including animation data, characters and textures are released under the Creative Commons Attribution License. It was rendered on Sun Microsystems' grid computing facility Sun Grid.

Fig. 8: Screenshot of the user interface.

The *Grabbable* dropped in one of the *Dropzones* is always annotated (Figure 9): 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 JSE, described in Section VI, more effective and precise by using better filter options. For example, if the user is looking for detailed information about a building's 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 [51][52][53] in usage in numerous multimodal dialog systems [48]. At the bottom of the graphical user interface, the user can either choose one of the generated Grabbables (Figure 8) or switch to the traditional Electronic Program Guide (EPG) view.

This approach breaks with the philosophy of TV apps where each app is linked to its own and single service. In this implementation, the attached JSE, is able to integrate different Web services, like Wikipedia, DBpedia, Freebase [54], Linked Movie Database (LinkedMDB) [55], 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 10, where combined personal data, such as zodiac sign or portrait pictures of DBpedia and Flickr, is shown as part of the biography. In

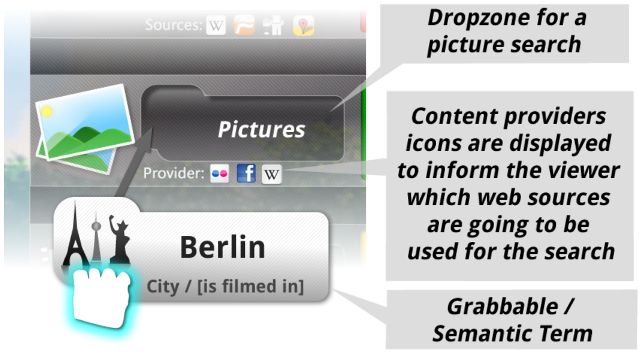
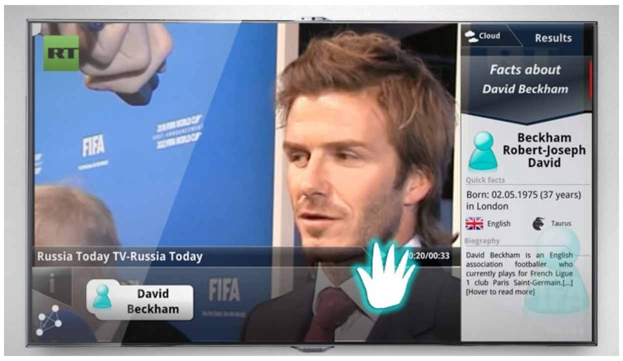

Sources:
Pictures
Provider:
Berlin
City / [is filmed in]
Dropzone for a picture search
Content providers icons are displayed to inform the viewer which web sources are going to be used for the search
Grabbable / Semantic Term

Fig. 9: Close-up of a *Dropzone*.

Figure 10, detailed facts about the famous football player "David Beckham" are displayed on the right side of the user's interface.



Beckham Robert-Joseph David

Quick facts

Born: 02.05.1975 (37 years) in London

English Taurus

Biography

David Beckham is an English association footballer who currently plays for French Ligue 1 club Paris Saint-Germain.[...] [Hover to read more]

Fig. 10: Display of David Beckham's biography.

Figure 11 shows the results of a search for pictures that was 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)

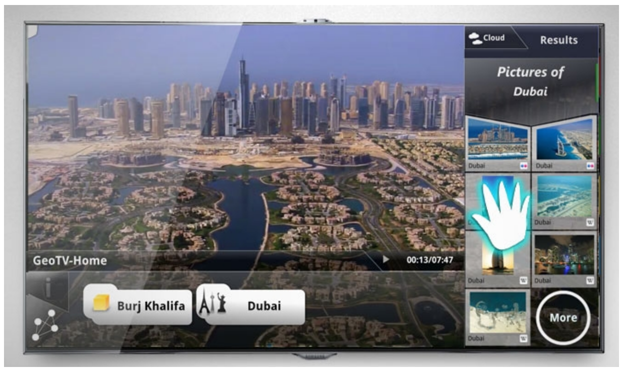


Fig. 11: Picture request during a report about Dubai with results coming from different Web sources.

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 12). 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 [56].

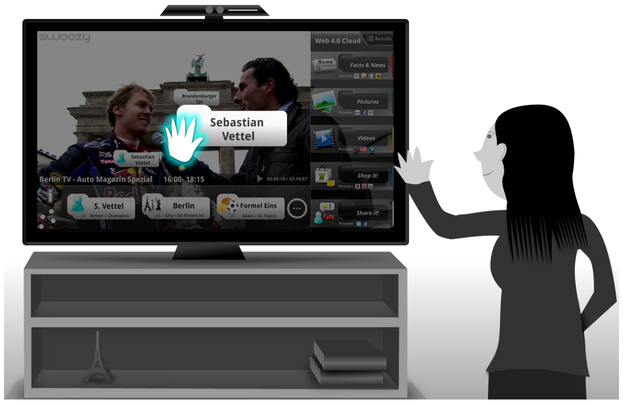


Fig. 12: User gesture interaction: a virtual hand allows the user to grab out a semantic term from a video.

We have deliberately implemented only two gesture types: *Grab'n'drop* and the *Push*-gesture, as these interactions are simple to realize, 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 13 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.

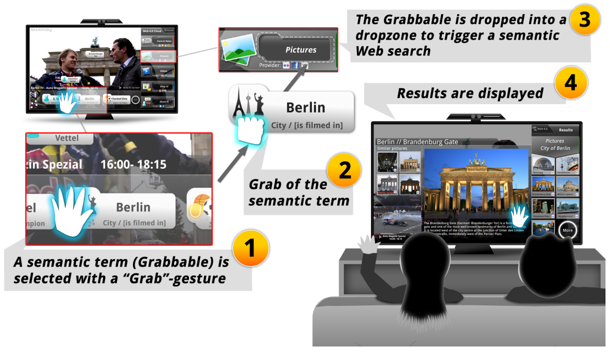

The Grabbable is dropped into a dropzone to trigger a semantic Web search
3
Results are displayed
4
Pictures
Berlin
City / [is filmed in]
Grab of the semantic term
2
Vettel
in Spezial 16:00- 18:15
Berlin
A semantic term (Grabbable) is selected with a "Grab"-gesture
1
Berlin // Brandenburg Gate

Fig. 13: Grab'n'drop interaction steps to trigger a picture search for the city of Berlin.

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, videos or even shopping recommendations during his favorite TV program.

D. Mobile client application

In some cases and especially when several viewers are watching TV together, it is necessary to let them look for information without interrupting or disturbing the main television "screen". With the mobile application of our approach, depicted in Figure 14 - the mobile Swoozy App (for Android and iOS) - 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. [49], where multimedia content is transferred from mobile devices to a kiosk system.

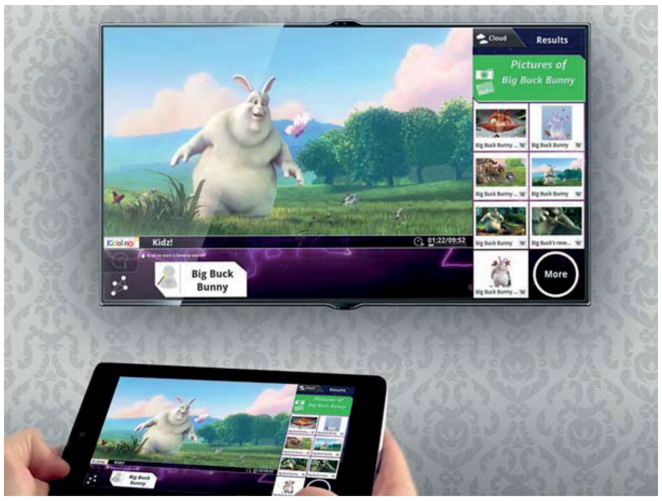


Fig. 14: Swoozy - mobile client application.

VI. Retrieval of Factual Knowledge Based on Semantic Technologies

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

A. Motivation

As mentioned at the beginning of this article, 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. Via these prepared input structures out of the extraction processes, the viewers are able to trigger queries to conventional Web services or Semantic Web Services over simple gesture interaction without the need of special skills, such as programming Web service APIs or the need to learn specific database query languages like RDF(S) or query languages like SPARQL [57][58]. For non-specialists it would be very hard to formulate such queries. Indeed, these query languages are primarily used to access the full power of the Semantic Web, by allowing a navigation through semantically annotated data sets by enabling and simplifying the search for specific instances corresponding to a given request.

We assume that the typical viewer does not really want to explicitly formulate his search queries 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 performs a judicious orchestration and mashing up of Web 2.0 and Semantic Web services and provides aggregated results coming from several sources - in this case Web services - as a final result. All retrieved results are returned to the client and displayed on the respective user interfaces (the television UI and/or the 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 15 shows an overview of the architecture design of the specific JSE backend component adjusted for the Swoozy domain. The JSE is composed of several modules, the *Query and Presentation Manager*, the *Planning Engine*, the *Execution Engine*, the *Context Broker*, the *Mapping Core*, and the *Semantic Service Repository*, which are described in the following subsections according to the components processing workflow.

C. Query processing

The “Query” module of the *Service Engine* [59] retrieves and decomposes the user’s query. The produced query structures

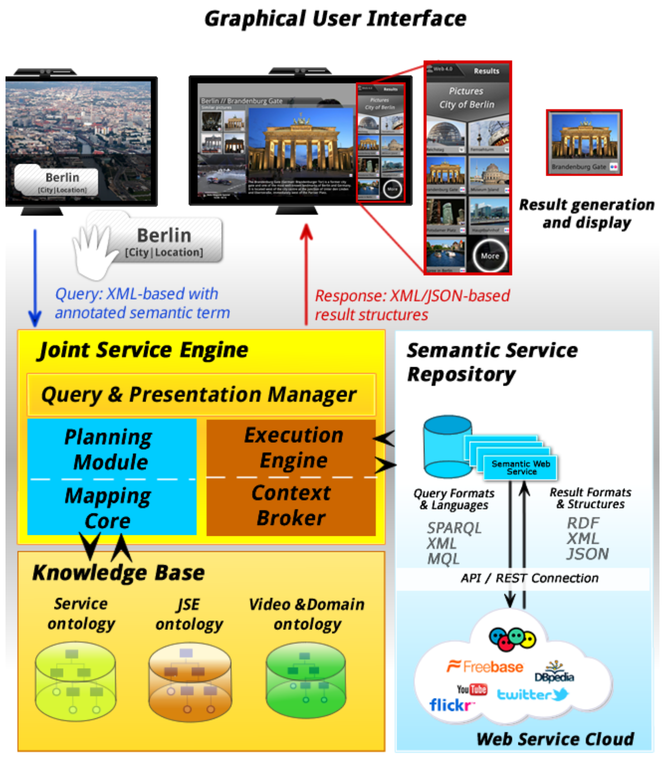


Fig. 15: Architecture of the Joint Service Engine.

are formulated according to a terminology defined by domain ontologies. They characterize the created scenery and the queries are formulated according to current generic template-based query structures, as shown in Figure 16. Each individual decomposed query part is mapped to a local meta-representation, the JSE ontology, modeled in OWL [11]. The ontology serves the purpose of precise definition of the current domain knowledge, the correct description of the retrieved content or data using a constructed reliable model based on the present situation and the environment. According to the user’s query, basic ontological components like individuals are created. Based on the defined vocabulary of the JSE ontology, the planning module looks in a hard matching process 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 types 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

```
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}
```

Fig. 16: Query search topics and properties.

calls which must be executed is 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 the 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. Context-based brokerage

The component for context-based processing is named *Context Broker*, and is responsible for the baseline analysis of the context, prediction, and the derivation of facts. Due to its flexible and generic nature it is easily integrable into the existing workflow of the JSE. It handles the concrete extraction of the services and gathers all information and result structures. The context-dependent filtering, evaluation and explanation of data structures prevents the use of outdated or inappropriate data, and thus ensures the correct integration of knowledge structures.

The process regarding the data fusion and interpretation is executed in the following three stages:

- Merging data structures and fusion of data sources on the basis of each present context - defined by the client (*Data fusion*).
- Evaluation and transfer of information to the internal knowledge representation of the Swoozy domain - JSE Ontology (*Interpretation of data*).
- Provisioning of harmonized information that can be accessed by the *Output Presentation Manager* (*Data deployment*).

The JSE provides service functionality and access to specific data by linking community-specific data sources. As a result, this component harmonizes and consolidates the information, which is provided by several heterogeneous services like DBpedia, Freebase, Linked Movie Database or Wikidata. The *Context Broker* component distinguishes between personalization rules, mapping rules and filters. On the lowest level of complexity, there are dynamically extensible filters that check for keywords. If during a process, a filter like the blacklist filter matches, which ensures children friendly results, the data are not being taken into account at all for further processing. Figure 17 shows the *Context Broker* and the multi-stage process of filtering that forms the essential task of this component. The functionality of filter-based rules and simple filter routines is used in different processing stages of the JSE: integration, interpretation, mapping and fusion of data. The goal is, to be able to do an absolutely coherent and correct mapping of the retrieved contents to embed heterogeneous external data structures into the existing knowledge structure. A comprehensive and reliable retrieval and detection of information creates the added-value and improves the Swoozy approach.

Personalization also plays an important role within this component: based on the context description, individual context models can be defined. A rule for context modeling consists of an explicit ontological description to interpret and structure the present situation. The *mapping rules* are necessary in order to map the available contents to specific instances in the style of the internal ontology. First, for the analysis of the obtained data with the designated filters, the contents are bound in interim meta-instances and released as suitable concrete instances for the downstream modules at the end of the processing chain. All retrieved and fused data structures from the *Execution Engine* are assigned and mapped to meta-instances of the JSE-ontology by predefined patterns of mapping rules and added to a temporary list. Each mapping rule contains implicit expert knowledge, enriched and expanded with the user's personal information, which defines, how to map the connected external data sources to the internal representation, and how to solve potential conflicts between different application domains. Depending on the query, it might be necessary, to merge the meta-instances of the temporary list or to directly create links between the present meta-instances. Within the preceding orchestration process one result - for example the location where an actor was born - can be used as input for the next service, to get extended location information like longitude and latitude of the place of birth. Before the results of the temporary list can be processed internally, they are subjected to a reconsideration by special filter-based rules. This processing step is needed to filter out duplicates, to merge individual properties and to apply information extraction methods. Information extraction means the retrieval of structured information from the present texts by methods of the field of Natural Language Processing (NLP), e.g., Named Entity Recognition is used for the extraction of facts and automated tagging of contents with topics like names of cities, places or locations, names of persons or buildings [40][60], as previously described in Section II-F. In this final step, the remaining instances from the temporary list are provided for further processing and the context broker will set up and draw new relationships between the retrieved extracted facts of the instances. This is done based on predefined patterns for persons, monuments, objects and fictive characters, e.g., *person X isBornIn city Y*.

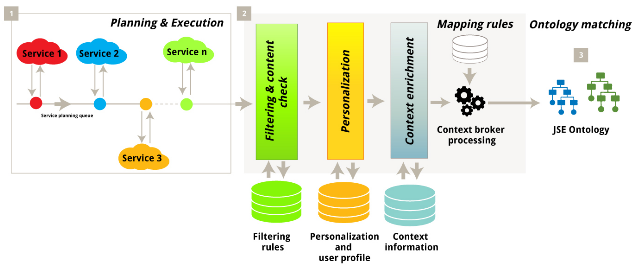


Fig. 17: Context Broker component, filtering and mapping mechanisms.

Now, after all the filters and upstream modules were introduced a concrete example will give a better overview of the internal process workflow of the module. The internal process workflow, illustrated in detail in Figure 18, is controlled by the central planning module and begins with an input structure that is formulated as a user query, e.g., “Show me more pictures of an actor!”. After the query processing the matchmaking process looks for adequate services for the decomposed query parts. As a result all present services are listed, even orchestrated services can be included. Each service gets called by the execution engine and in a preprocessing step, the incoming result structures are screened for several keywords and string characters, defined in an extendable list of multiple rules. In cases of failures (e.g., when the service is not reachable) the component will monitor it and will reschedule the search task by creating a new plan using alternate services. Content that passed the initial filtering is mapped to instances based on the internal ontology and on predefined mapping rules that exist for each service. These mapping structures form the so-called external expertise to trigger the mapping of the current data to the local domain. In an advanced filtering process, the instances are analyzed to find duplicates: the results are then weighted based on named entity recognition, personalization rules, and in a final step relations are drawn between the resulting items. All remaining harmonized result structures are linked to concrete instances that are then forming the ontological final result representation.

In the deployment stage, the *Context Broker* component prepares the internal OWL-Structure for output presentation. Customer or user’s personal information and the demanded output format or view of the output structures are deposited in the JSE-Ontology. It is then passed as is to the *Presentation Manager* to create specific output structures in a REST-client friendly format, like XML or JSON.

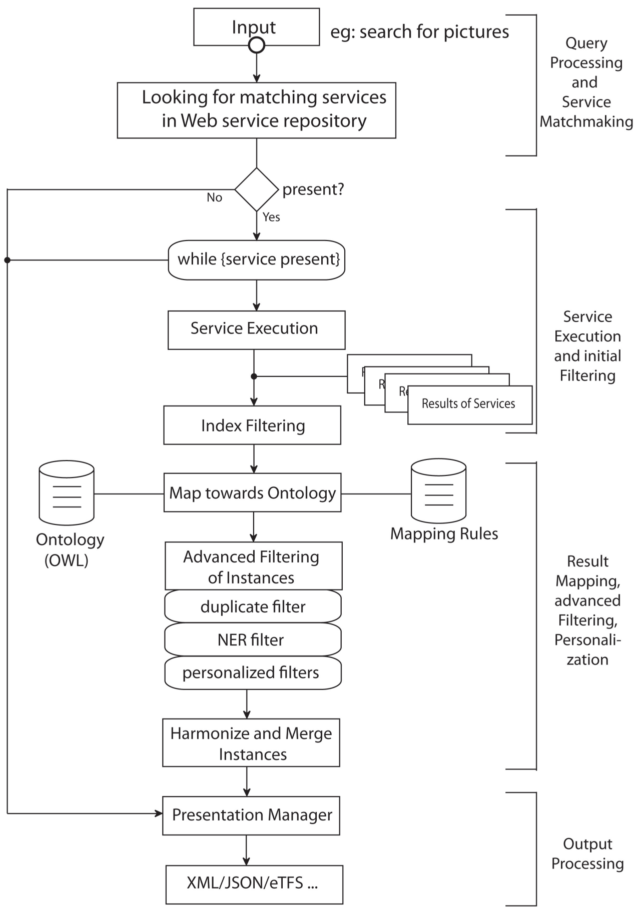


Fig. 18: Context Broker processing workflow.

E. Mapping and matching

During the processing chain inside 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 during 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 range of result formats varies from simple JSON structures to complex semantic data structures like RDF. 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 their internal data structures.

F. Semantic 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 a prototypical implementation [59] concepts for detailed service descriptions in OWL-S [18] are created and deployed in this *Semantic Service Repository*. Concrete service descriptions are realized for freely available knowledge sources, such as DBpedia, Freebase, Flickr or Linked Movie Databases. In these Web-based systems, information is stored and made accessible in a structured and manageable form, which would be otherwise difficult to access through 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 [61]. The JSE also 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 according to a well-defined domain ontology.

The major part on the technical side of the *Semantic Service Repository* is the discovery and matchmaking of services. The repository hosts, provides and manages descriptions of various pre-annotated sources based on Semantic Web technologies. This component provides more flexibility through simple modifications, like an add-, delete- and replace-functionality of stored service descriptions, and uses a modular design for individual components without stringent programming dependencies. Inspired by the approaches of Sirin [21] and Lambert and Domingue [23], all Web services in our *Semantic Service Repository* are represented in OWL-S with a grounding declaration in WSDL [62] or WADL [63].

The operation of the internal discovery process of the *Semantic Service Repository* was described in detail in the description of the generic flexible framework [61]. Figure 19 presents a simplified overview of the internal processing steps. A *Broker* consists of a *Query Handler* that interprets a query from the *Planning and Execution* module. The query is formulated in XML based on a single XML schema. From this interpretation, facts derive abstract types of services. By means of a *Rule Engine*, SPARQL queries can be derived from the decomposed query facts, which are used for the matchmaking process. This query is then forwarded to the *Matcher*, that connects to the service repository and executes the received SPARQL query, which points to adequate and concrete Semantic Web Service representations of deposited services. If the query matches, a list of semantically described services is returned. The used service ontology describes how to interpret and execute the external service. All required parameters for the concrete execution of the service and its external call are stored in this ontological description. The orchestration of services, their specific requirements and dependencies are also stored within this ontological description. In the latter, the grounding description of the external services is based on the technical concepts of WADL and WSDL. These are describing the technical aspects of communication and have been extended within our system with parameters and properties that refer to pattern generated SPARQL queries. Each SPARQL query also contains parameters used within the output structures. Those parameters have a direct mapping to the internal ontology and they are mainly used to call specific knowledge databases like DBpedia or triple stores.

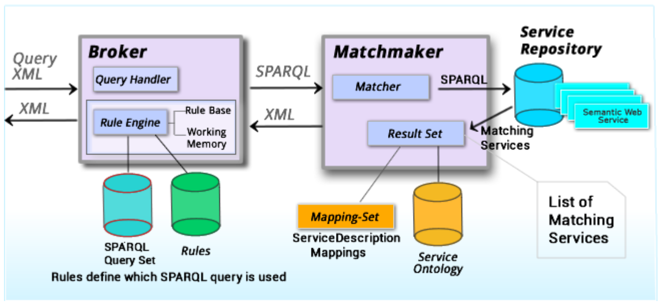


Fig. 19: Service discovery process.

A combined result structure is passed to the *Planning and Execution* module, which triggers the process of integration, analysis and harmonization of external data structures.

G. 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 parsing process. Depending on the user's query, e.g., a media search, different structured output formats (RDF, XML, JSON, etc.) might 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 screens and television systems are fed with the results coming from the *Presentation Manager*. With this parallel output processing a cross-media interaction is possible.

H. Data and semantic service management

The choice of which Web service is going to be internally triggered is based on rules and filters as defined in Section VI-E . These rules are not static and can be changed over time by a dedicated Web-based management board: there it is possible to adapt the queries and edit the underlying ontologies and rules used within the query process.

Moreover, via this management module, the ontology can be modified and defined mapping rules can be easily adapted: new

Web services and knowledge databases can be deployed, or activated. This aspect also leverages the modular and distributed cloud-based concept of our system: each service is easily interchangeable due to their generic ontology-based definition. The possibilities to edit rules and to select the required Web services are motivated by the pure semantic orientated approach of our system. Additionally, in parallel to free accessible internet based content, like DBPedia, it is possible to adapt the system in accordance to the broadcaster's need, to promote and give their viewers access to own premium content services (specific videos, interviews, infographics). Over the Web-based management board broadcasters can easily and quickly adapt the offered and displayed contents within the Swoozy UIs.

VII. Conclusion and future work

We demonstrate with our approach - the *Swoozy* system - and its prototypical implementation, that it is possible to provide a novel way to interact with video contents without interaction breaches. Through a seamless combination of gesture-based interaction, video information coming directly from the broadcasted signal, and the Joint Service Engine as backend service connector it is possible to enhance the displayed content with additional information from the Internet (Web or Semantic Web) and its related services. The scenery is underpinned by additional information of external services, wherein this information is context-based, machine-readable and interpretable.

An extended version of the Swoozy system is currently in usage in several living labs in Germany and connected to third-party multimedia platforms targeted to various application domains: the system has also been presented to a wider audience during international exhibition fairs, e.g., at the CeBIT, and there it generated a lot of positive user feedback, especially the innovative exploration in connected knowledge databases.

Moreover, thanks to this innovative approach, television enters into a new dimension in which viewers will receive additional information and knowledge about the persons, locations, and objects featured in their favorite television programs.

The *Swoozy* concept offers a wide range of capabilities and possibilities to communicate and is not only applicable for the sole field of television. The concept can also be used to enhance the functionality of existing video-based systems, such as video-on-demand platforms, 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] M. Deru and S. Bergweiler, "Swoozy - An Innovative Design of a Distributed and Gesture-based Semantic Television System," in Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014), International Academy, Research, and Industry Association (IARIA). IARIA, 8 2014, pp. 131–139, best Paper Award.

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

[3] 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.

[4] 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.

[5] 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.

[6] B. Makni, S. Dietze, and J. Domingue, "Towards semantic TV services a hybrid Semantic Web Services approach," 2010, [Retrieved: May 2015].

[7] 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

[8] 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/

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

[10] 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/

[11] P. F. Patel-Schneider, P. Hayes, and I. Horrocks, "OWL Web Ontology Language Semantics and Abstract Syntax," W3C Recommendation, 2004, [retrieved: May 2015]. [Online]. Available: http://www.w3.org/TR/2004/REC-owl-semantics-20040210/

[12] 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.

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

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

[15] Institut für Rundfunktechnik, "Broadcast Metadata Exchange Format," BMF 2.0, 2012, [retrieved: May 2015]. [Online]. Available: http://bmf.irt.de/

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

[17] 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.

[18] D. Martin et al., "OWL-S: Semantic Markup for Web Services," W3C Submission, 2004, [retrieved: May 2015]. [Online]. Available: http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/

[19] D. L. McGuinness and F. van Harmelen, "OWL Web Ontology Language Overview," W3C Recommendation, 2004, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/owl-features

[20] E. Sirin, B. Parsia, and J. Hendler, "Filtering and Selecting Semantic Web services with Interactive Composition techniques," IEEE Intelligent Systems, vol. 19, no. 4, 2004, pp. 42–49.

[21] E. Sirin, "Combining Description Logic Reasoning with AI Planning for Composition of Web Services," dissertation, pp. 1–239, 2006.

[22] O. F. F. Filho and M. A. G. V. Ferreira, "Semantic Web Services: A RESTful Approach," in IADIS International Conference WWW/Internet. IADIS, 2009, pp. 169–180, [retrieved: May 2015]. [Online]. Available: http://fullsemanticweb.com/paper/ICWI.pdf

[23] D. Lambert and J. Domingue, "Grounding semantic web services with rules," in Proceedings of the 5th Workshop on Semantic Web Applications and Perspectives (SWAP), ser. CEUR Workshop Proceedings, A. Gangemi, J. Keizer, V. Presutti, and H. Stoermer,

Eds., vol. 426. CEUR-WS.org, 2008, [retrieved: May 2015]. [Online]. Available: http://ceur-ws.org/Vol-426/swap2008_submission_8.pdf

[24] 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.

[25] 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.

[26] 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.

[27] 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.

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

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

[30] 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.

[31] 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.

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

[33] 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.

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

[35] 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 (ESWC'2005). Citeseer, 2005.

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

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

[38] K. Merkel, "HbbTV - Status und Ausblick," 2012, [retrieved: May 2015]. [Online]. Available: http://www.irt.de/webarchiv/showdoc.php?z=NTgwNyMxMDA1MjE1I3BkZg==

[39] 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. Association for Computational Linguistics, 2007, pp. 36–41.

[40] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005, pp. 363–370.

[41] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives, "DBpedia: A Nucleus for a Web of Open Data," in The Semantic Web, ser. Lecture Notes in Computer Science, K. Aberer et al., Eds. Springer Berlin Heidelberg, 2007, vol. 4825, pp. 722–735.

[42] I. Smart TV Alliance, "Technical Specification Version 4.0," 2014, [retrieved: May 2015]. [Online]. Available: https://sdk.smarttv-alliance.org/download.php?file=Smart_TV_Alliance_v4.0_specification.pdf

[43] "HbbTV 2.0 Specification," 2015, [retrieved: May 2015]. [Online]. Available: https://www.hbbtv.org/pages/about_hbbtv/HbbTV_specification_2_0.pdf

[44] 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

[45] Samsung, "Design Principles for Creating Samsung Apps Content," 2013, [retrieved: May 2015]. [Online]. Available: http://www.samsungdforum.com/UxGuide/2013/01_design_principles_for_creating_samsung_apps_content.html

[46] 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.

[47] 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.

[48] 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.

[49] T. Becker, M. Löckelt, C. H. Schulz, S. Bergweiler, M. Deru, and N. Reithinger, "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.

[50] 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.

[51] 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.

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

[53] 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.

[54] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in Proceedings of the 2008 ACM SIGMOD international conference on Management of data. New York, NY, USA: ACM, 2008, pp. 1247–1250.

[55] O. Hassanzadeh and M. P. Consens, "Linked Movie Data Base," in Proceedings of the Workshop on Linked Data on the Web, LDOW, 2009, [retrieved: May 2015]. [Online]. Available: http://ceur-ws.org/Vol-538/ldow2009_paper12.pdf

[56] 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.

[57] "SPARQL query language for RDF," W3C Recommendation, 2008, [retrieved: May 2015]. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[58] "SPARQL 1.1 query language," W3C Recommendation, 2013, [retrieved: May 2015]. [Online]. Available: http://www.w3.org/TR/rdf-sparql-query/

[59] 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.

[60] G. Neumann, G. Paaß, and D. van den Akker, "Linguistics to structure unstructured information," in Towards the Internet of Services: The THESEUS Research Program. Springer, 2014, pp. 383–392.

[61] S. Bergweiler, "A Flexible Framework for Adaptive Knowledge Retrieval and Fusion for Kiosk Systems and Mobile Clients," in Eighth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2014), International Academy, Research, and Industry Association (IARIA). IARIA, 8 2014, pp. 164–171.

[62] E. Christensen, F. Curbera, G. Meredith, and S. Weerawarana, "Web Services Description Language (WSDL) 1.1," W3C Note, 2001, [retrieved: July 2014]. [Online]. Available: http://www.w3.org/TR/wsdl

[63] M. Hadley, "Web Application Description Language (WADL)," W3C Member Submission, 2009, [retrieved: May 2015]. [Online]. Available: http://www.w3.org/Submission/wadl