

# MediaCaddy - Semantic Web Based On-Demand Content Navigation System for Entertainment

Shishir Garg, Amit Goswami, Jérémy Huylebroeck,  
Senthil Jaganathan, and Pramila Mullan

France Telecom R&D, 801 Gateway Boulevard, Suite 500, South San Francisco, CA, USA  
{shishir.garg, amit.goswami, jeremy.huylebroeck,  
senthil.jagnathan, pramila.mullan}@rd.francetelecom.com

**Abstract.** This paper is aimed at documenting the role of Web services and specifically Semantic Web Services in serving the needs of the entertainment industry by enabling the users to easily research and explore the large volume meta-content (content about content e.g. entertainment news, articles, reviews, interviews, trailers etc) and eventually leading them to FIND right content (Music, Movies TV program etc). In this scenario, semantic web techniques are used to not only develop and populate the ontology from different meta-content sources, but also to annotate them semantically to provide personalize meta content based Search-Find experience for main content. The paper outlines an application scenario where this is applied in a demonstrated proof of concept and articulates the next steps in the evolution of this architecture.

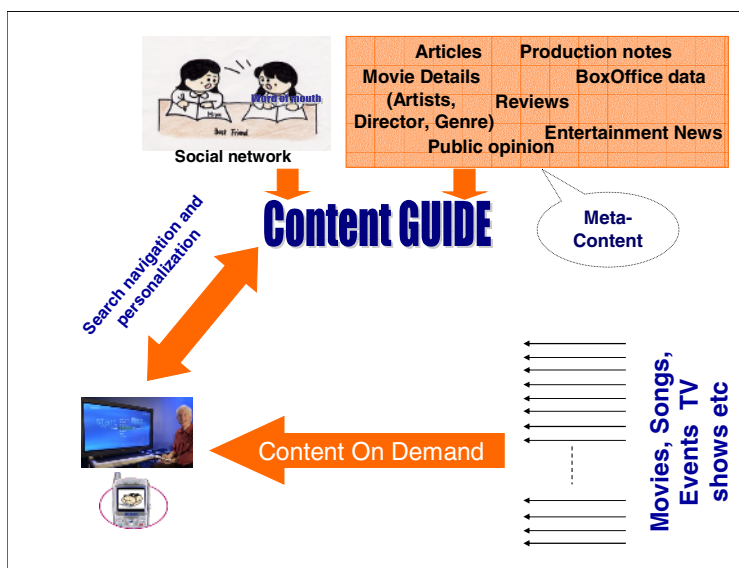
## 1 Overview of the Entertainment Industry

For the entertainment industry, traditional approaches to delivering meta-content about movies, music, TV shows, etc. were through reviews, articles etc that were done and published in traditional media such as newspapers, magazines, TV shows, etc. These entertainment magazine/new papers have provided extensive Subjective analysis of Movies, TV shows, Events and music, hence played a very critical role in influencing consumer's opinion about different types of contents. In other words, Consumer has used these sources to search and find right content for him/her.

With the introduction of the internet, non-traditional forms of delivering entertainment started surfacing. The third quarter of 2003 in the U.S was the best ever for broadband penetration bringing such services as content on-demand and mobile multimedia, back from hibernation. 2003 also witnessed the beginning of the on-line content explosion and a flood of new, more capable terminals. Users are accessing content via their television set top boxes, ISP portals, and smart phones, interactively. As of today more than 5000 movies and 2,500,000 songs are available online. In the next couple of years this figure is expected to grow in leaps and bounds. With such a phenomenal rise in content over IP, a new need for secondary meta-content related to the movies/music emerged. Initially this was through movie reviews or music reviews, etc. published on web portals such as Yahoo, MSN, etc and online magazine portals as well as entertainment sales sites such as Netflix.com. Amazon.com etc... Most Consumers today get information about media content

primarily from reviews/articles in entertainment/news magazines, their social network of friends (one user recommends a song or movie to a friend) and acquaintances, and advertisements. In most of the cases, one or all of the above influence user's opinion about any content she chooses to consume. In addition, a new breed of customizable meta-content portals has emerged, which specifically targets for the entertainment industry. Examples of such portals include, Rotten Tomatoes, IMDB, etc. However, these services today are typically accessed via portals thereby limiting the interactions and access to the information to exchanges between a user and the source for non-PC environment.

Today, Content on demand is finally taking off due to boost in digital TV infrastructure and added intelligence in Set Top Boxes—by 2008 more than 40 million set top boxes will have advanced OS and hardware. Interactive Program Guides (IPGs) are becoming the entertainment portal for TV viewers, and In-Stat/MDR expects the worldwide IPG market value to grow to nearly \$1 Billion (US) by 2008. IPGs will help end-users to find a TV program, movie or sporting event from among listings of thousands of available options, and then make it easy for them to select the program for viewing, tag it with a reminder for later, or even set up a recording to capture the show for time-shifting on a Personal Video Recorder.



**Fig. 1.** Conceptual Model of Content Navigation System

This paper discusses a service concept for a recommendation engine called MediaCaddy which leverages semantic web technology to build intelligence in IP based content portals for any connected device. It is a recommendation and aggregation service built around a self-learning engine, which analyzes a click stream generated by user's interaction and actions with meta-content displayed on UI. This meta-content (Music /Movies/ TV reviews/ article/ synopsis/ production notes) is accessed from multiple Internet sources and structured as an Ontology using a

semantic inferencing platform. This provides multiple benefits, both allowing for a uniform mechanism for aggregating disparate sources of content, and on the other hand, also allowing for complex queries to be executed in a timely and accurate manner. The platform allows this information to be accessed via Web Services APIs, making integration simpler with multiple devices and UI formats. Another feature that sets MediaCaddy apart is its ability to achieve a high level of personalization by analyzing content consumption behavior in user's personal Movie/Music Domain and his social network and using this information to generate music and movie recommendations. Fig 1 explains conceptual model of MediaCaddy.

## 2 Service Opportunities

It has been proven from business cases of Internet e-commerce vendor Amazon and DVD rental portal Netflix [1] as well as TV Guide/PVR service Tivo [2] that simple user assistance/ recommendation engines can drive revenues up by 30% and 60% respectively. With the explosion of on-demand content, more intelligent methods can be developed to help a user navigate through a content reservoir using a limited device such as a remote control. Such an intelligent system will help in increasing content consumption via an on-demand application. Traditionally 80% of the entertainment industry revenues come from 20% of the content produced. In a typical revenue share model, users pay \$2 to \$4 per film, wherein the content owner takes 30% to 60% of it and the rest goes to a service provider. The more popular the content, the larger the content owner's revenue share. MediaCaddy thus can be used to enable an increase in consumption of lesser known content, where margins are higher, hence increasing the service provider's revenue share.

Let us look at some real life scenarios. These scenarios explain a user's interaction with the MediaCaddy enabled content guide.

Assumptions:

- At least three thousand content titles are available for user to consume.
- The system knows a user's content purchase history
- Every IM buddy of the user has a MediaCaddy enabled VoD service.

### 2.1 Pretext: (Traditional Content Guide)

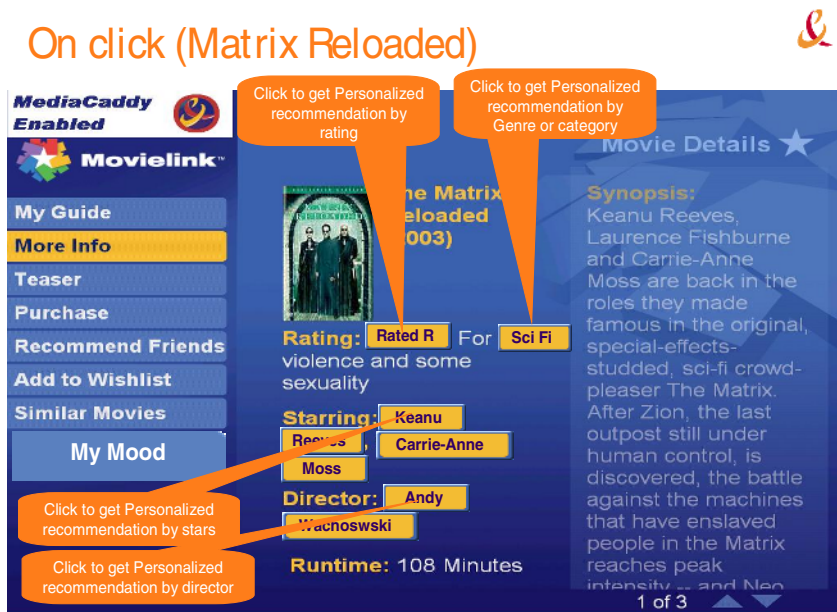
User is exploring content catalogue offered by his/her on-Demand service provider. On Demand System doesn't know anything about user, hence offers list of content categorized either by artists (approximately 5000+) or by Directors (approx 2000+) or by Genres (100+).

### 2.2 Scenario #1: (Content navigation Guide powered by MediaCaddy)

Along with traditional categories (Sort by name or Sort by Genre) offered by service providers, MediaCaddy offers three more categories. Each of these categories is explained in more detail.

The first category is a single comprehensive listing of recommended content. These movies/music tracks are suggested based on past purchase history of user. This

personalized list of is generated based on Components (Artists, directors, Genre, basic keyword based content description, MPAA rating, reviewers which had facilitated past purchases etc) of movies/music tracks purchased by the user in the past.



**Fig. 2.** Sample User Interface of Content Navigation System

The second category is a listing of content from a user's friend circle. This category will offer content recommended by the user's active friends (compiled from messenger buddy list) and movies liked by passive buddies (people of similar interests, compiled from comprehensive listing of "friends of friends")

The third category is a listing of content based on the user's current mood, generated based on interest shown by the user in the current session. The user's interest is captured by analyzing the clicks she made on any of the ontological entities displayed in the UI e.g. artist, genre, movie/music, producer, etc. (see Fig. 2). For example, a user comes home and is interested in watching an action movie by either Tom Cruise or by Tom Hanks. The movie should also be either a spy thriller or a super natural thriller. She starts browsing the catalogue, and based on her interests captured by following and analyzing clicks made in the current session, a list of movies is offered that fits her interest.

### 2.3 Scenario #2: (Content navigation Guide powered by MediaCaddy)

Once the user selects a movie from one of these four sources (system recommendations, social network, mood based or from catalogue), she can get more detailed information such as artists, director, category, synopsis, personalized reviews by her preferred reviewers (generated based on past performance of reviewers in influencing user's past purchases), recent news items from preferred news sources, productions notes, box office results, public consensus etc.

### 2.4 Scenario #3: (Content navigation Guide powered by MediaCaddy)

Upon viewing the details of a selected movie, if the user is not interested in it, she can navigate further using any of following mechanisms:

- The user can request a personalized content list by an artist by clicking on the artist's name from the current screen.
- The user can request a personalized content list by a director by clicking on the director's name from the current screen.
- The user can request a personalized content list by a genre/category by clicking on the genre/category's from current screen.
- The user can request a personalized list of movies by MPAA rating by clicking on the MPAA rating from current screen.
- The user can request a personalized list of movies similar to a particular movie by clicking on it. This holds true for other content types as well.

### 2.5 Scenario #4: (Content navigation Guide powered by MediaCaddy)

Once a personalized list of movies is displayed, the user can also request the latest news either for the selected artist or director. The user will be offered news items from her preferred news sources.

### 2.6 Scenario #5: (Content navigation Guide powered by MediaCaddy)

Once the user consumes the purchased content, she will be given an option to rate that content on a scale of one to five. She can also recommend content to her active buddies. User participation into the system is encouraged in this manner.

## 3 Implementation Architecture

The following figure provides an overview of the implementation architecture used for MediaCaddy development. The development was done completely in a Java and J2EE environment, using an open source application server called JOnAS from ObjectWeb.org. The components include:

- Automated meta-content gathering engine: The meta-content gathering engine is a series of extraction agents, built as a Classifier Committee, using a combination of statistical, probabilistic and knowledge based data to populate the ontology from semi-structured and non-structured data sources. Trusted partner data, available in structured form is populated into the ontology server directly.
- Meta-content repository: basically refers to the ontology server (containing an ontology representation as well as instance data), which is based on off the shelf technology from a third party, and is a core asset used in producing recommendations. A REST based Web service API is available to browse the ontology and run inference queries on it. The technology is described here [3]
- Social Network components: These are built using EJBs for the core server with business logic built in plain java objects, and is based on a Cosine similarity algorithm. The cosine algorithm simply computes the cosine value between two

vectors of any dimension. The space considered is orthogonal in this implementation. These components are not detailed in this paper.

- **MediaCaddy Server:** This is the heart of the system. The MediaCaddy server performs data transformations across various data types and formats, by defining a system wide standard XML format that all components should adhere to. The MediaCaddy server is also interfaces with other components of the system including user profiling, session management and impulse generation, social network, etc. Naturally, the MediaCaddy server also interfaces with the Ontology server via the query interfaces provided by the ontology server. For this, the MediaCaddy server also defines a Query Management API to standardize the interactions with the ontology server.

Scalability issues discussed in previous projects such as [4] are not addressed.

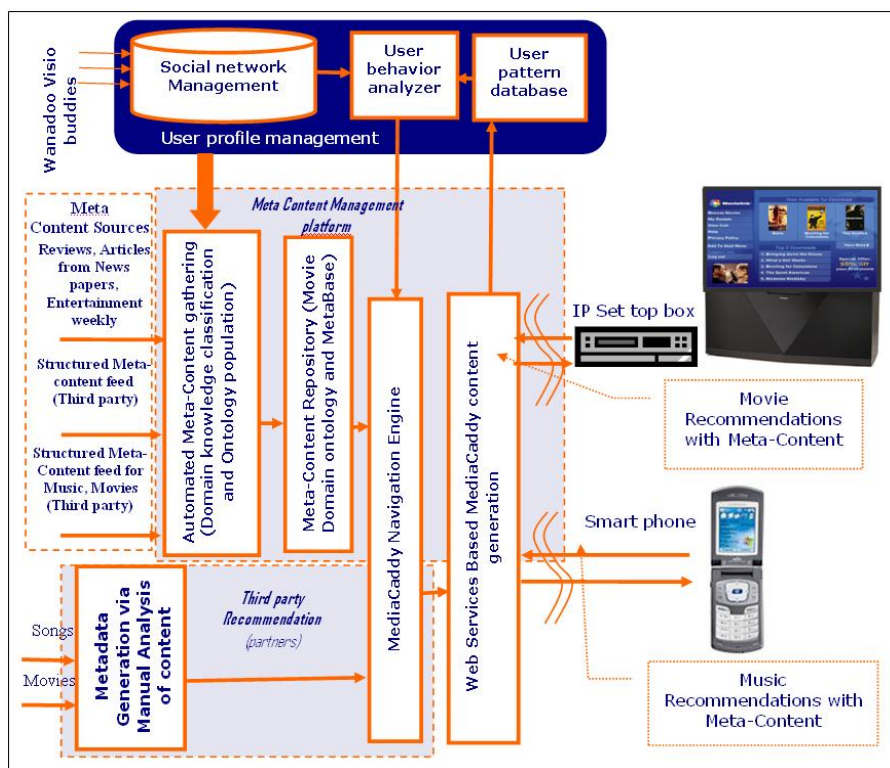


Fig. 3. MediaCaddy component level view

#### 4 How Semantic Web Technologies Address the Needs of the Entertainment Domain?

A Semantic web approach to this problem is interesting as the entertainment domain offers a vast set of resources that are available, often for free, to be leveraged in

developing a complex system such as MediaCaddy. In working with commercial third-party sources, we are able to use structured data to populate the ontology, while the ontology has been completely pre-designed by hand.

Using a non-semantic, simple XML and XML-Schema based system would limit the levels of flexibility we are able to achieve. While XML provides for well formed messaging, the more sophisticated Semantic web technologies provide support for complex querying and inferencing. Machine readability and automated processing, key tenets of the Semantic web world, are key requirements for the MediaCaddy platform. Every query made by the user is translated into several system level queries. It is not possible to cover the details of the syntax used within the system to describe the ontology in this paper.

By leveraging Web services for their ability to easily integrate multiple heterogeneous sources of meta-content and leveraging semantic web technologies to semantically annotate the content we are able to empower the users of entertainment related meta-content to be able to easily maneuver the wealth of entertainment-related information in order to find the information that they are specifically looking for. Furthermore, Web services based interfaces enable easier creation of applications that can also leverage this wealth of information in unique ways to present to users. One other by-product is the ability to seamlessly integrate multiple partners in the entertainment value chain.

The following four sections represent the main components of the MediaCaddy system.

## 5 Ontological Representation of Personalized Movie Domain

The subject of *ontology* is the study of the *categories* of things that exist or may exist in some domain. The product of such a study, called *ontology*, is a catalog of the types of things that are assumed to exist in a domain from the perspective of a person who uses a language for the purpose of talking about the domain. The types in the ontology represent the *predicates*, *word senses*, or *concept and relation types* of the language when used to discuss topics in the domain. An un-interpreted logic imposes no constraints on the subject matter or the way the subject may be characterized. By itself, logic says nothing about anything, but the combination of logic with an ontology provides a language that can express relationships about the entities in the domain of interest.

An ontology is an explicit formal specification for representation of objects, concepts and other entities that are assumed to exist in some area of interest and the relationships that hold among them. It is a specification of a conceptualization. Since the ontology defines concepts and relationships, traversing the ontology allows for discovery of relationships that exist between entities, thus enabling an ontology inference layer to return deterministic responses to queries.

In order to ensure success in this experiment, it was important to select a domain where the number of entities is stable and relatively small. Also, it is desirable to have sufficient existing information about the domain that needs to be modeled using the ontology. The entertainment domain fits these requirements well, and the movie domain specifically is extremely well suited with a lot of disparate yet structured

sources available that can be used not only as knowledge sources to model the ontology but also as sources to populate the entity instances.

The MediaCaddy ontology defines several entities as discussed in Figure 4. The main entities, the movie entity and the user entity, define the relationships that exist between these entities and several other entities. The relationships can be categorized into the following types:

- Direct entity-entity relationships: These are the various relationships such as movie-review, movie-director, movie-artist, etc. They are defined as explicit relationships that help ascertain the unique relationship between these entities. For example, the movie-director relationship is defined as “is directed by”, etc. There are several user related relationships as well, such as user-artist and user-reviewer, that are used extensively within MediaCaddy to personalize the system for the user’s needs
- There are a few entity-same entity relationships as well, that are meant to define relationships between similar instances. In the case of users, it is used to define affinity between individuals.
- Multi-hop relationships: These are used to traverse relationships between multiple entities and to infer details not obvious without traversing the defined links. For example, by traversing the user -> reviewer -> movie graph, additional movie instances are revealed than originally anticipated. Such traversals are key to the MediaCaddy system.

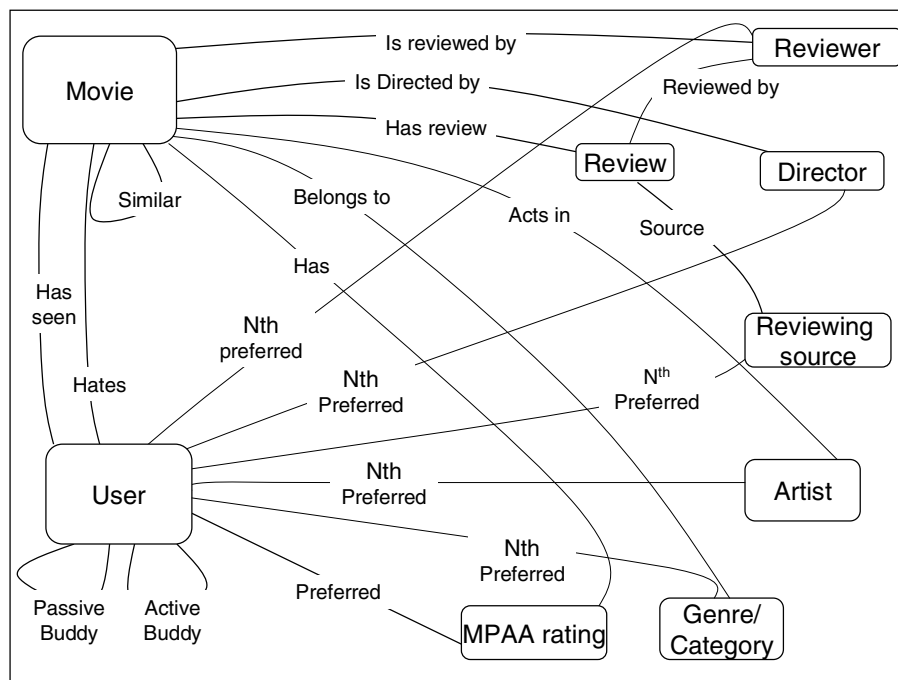


Fig. 4. Movie Ontology



Ontological relationships between entities are used to infer more information about a user profile than is seen explicitly from a user's behavior. Furthermore the system constantly captures user activity (Clicks made on UI e.g. Fig 2) which in turn updates the user profile. This approach distinguishes MediaCaddy's semantic ontology approach from traditional recommendation systems.

For a novice user the recommendation made by the system is of generic nature (default settings) but as the usage increases the recommendations are more accurate and ontological relationships helps in building a better user profile by inferring different types of possible relationships between entity instances. User preferences are updated on a timely basis based on usage patterns and the user profile is then built by correlating previous patterns with the current pattern. The profile is then reflected back in the ontology.

There are several additional pieces of information stored in the ontology, such as content related metadata (e.g. year of release, length, etc.) and other details that are not clearly exposed in the metadata described as they are stored as attributes of the entities defined.

## 6 Content Knowledge Capture (Ontology Building)

MediaCaddy utilizes a third party component to perform crawling, indexing and population of Ontological entities. The third party component crawls through different web based content information sources and extracts any movie related information. Once extracted, content component objects or entities (artist, director, etc.) are stored as ontology entity instances and relationships between different entity instances are established. A paper describing the components used can be found in [3].

## 7 User Profile Capture and Analysis

The Semantic clickstream capture and analysis system analyses the user's navigational and content selection inputs, stores them as impulses received from the user and processes them into semantic metadata about the end-user. This semantic metadata is stored in a domain specific ontology. The ontology represents the user's preferences within the domain specified in the ontology, for example, within a movie industry ontology the user's preferences could represent the interests the user has established based on past impulses and is used for various future interactions of the user.

The exchange of user data can take place using HTTP protocols, XML based protocols or Web services standards based exchanges.

The system defines a user clickstream analysis component that extracts relevant user impulse data and stores this data into a domain specific ontology. This is useful for subsequent interactions with the user that may be immediate or deferred.

The system defines four impulse types:

- Session related activities
- Selection impulses
- Content specific impulses
- Meta-data related impulses

These user impulses are then analyzed and aggregated and the user information is extracted into the ontology of the user.

## 8 Content Navigation Via Semantic Queries

### 8.1 Personalized Content Navigation in Movie Domain

MediaCaddy offers three different types of content navigation methods.

#### 8.1.1 User Profile Based Navigation

The user profile section explains how a detailed analysis of the user's behavior and interaction with content ontology, generates user profile, which is nothing but couple of lists of preferred directors, artists, news sources, review sources, reviewers, genre/category/keywords, active buddy and passive buddy. These lists will be used for personalization of navigation output. To understand personalized content navigation, we need to understand entry points in the Content Ontology domain. Please see Fig. 5 and Fig. 6 to understand flow of content navigation.

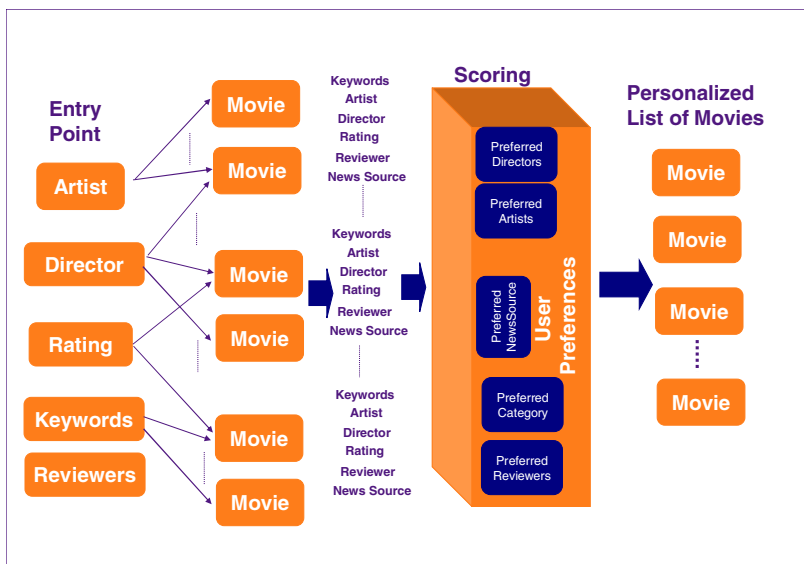


Fig. 5. Content navigation based on different entry points of Movie Domain

The following is a representation of personalized weight for every movie while performing navigation by artist, director or genre.

- (AW) = Weight of artist as a movie component in User's Movie Ontology
- (DW) = Weight of Director as a movie component in User's Movie Ontology
- (GW) = Weight of Genre as a movie component in User's Movie Ontology
- (RW) = Weight of Reviewer as a movie component in User's Movie Ontology

- (Ng) = Number of genres of movie
- (Na) = Number of Artists of movie
- (Nd) = Number of Directors of movie
- (Nr) = Number of Reviewers of movie
- (MW) = Personalized weight of a movie.
- N = Index of preferred artist, director, genre, reviewer, review sources and news source tables of user's profile. If entry is not found in table value of N will be zero.

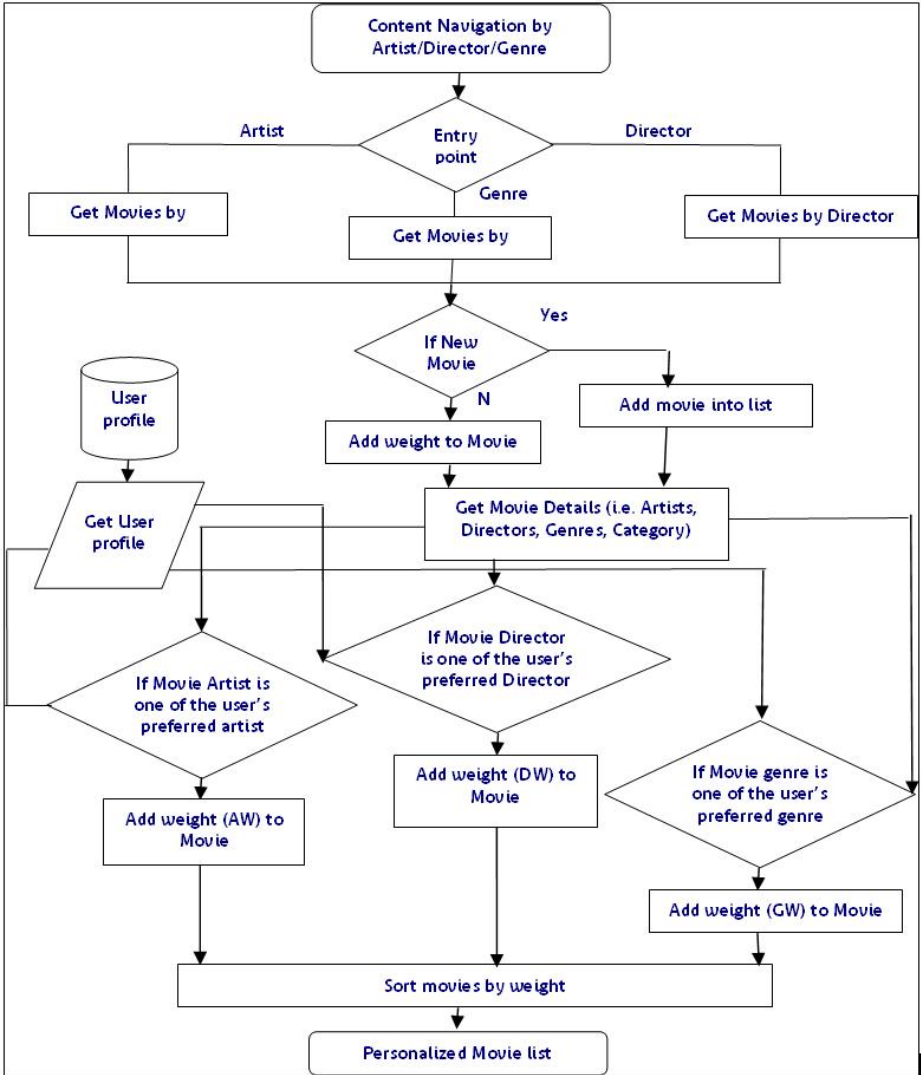


Fig. 6. Content navigation by Artist, Director or Genre

$$\text{Personalized Movie Weight (MW)} = \sum_{i=0}^{i=Na} (AWi / Ni) + \sum_{i=0}^{i=Nd} (DWi / Ni) + \sum_{i=0}^{i=Ng} (GWi / Ni) + \sum_{i=0}^{i=Nr} (RWi / Ni) \quad (1)$$

There are multiple Entry points in Movie knowledge domain graph. Typically one of the following will be used to request personalized movie listing.

1. Artists: User can request a personalized list of movies, just by clicking name of any of the artist from any of the guide's screen. Please refer 0 and 0.
2. Director: User can also request a personalized list of movies, just by clicking name of any of the director from any of the guide's screen. Please refer 0 and 0.
3. Genre/Keyword/Category: Third entry point in Movie knowledge domain can be genre/category/Keyword. User can request a personalized list of movies, just by clicking names of any of the category/keyword/Genre. Please refer 0 and 0.
4. User: user can also request a simple personalized list of movies based on his user profile.

### 8.1.2 Mood Based Content Navigation

Both of the above discussed content navigation methods heavily depend on user's profile, but its very difficult to profile human behavior, hence this one is totally independent of user profile. Mood based content navigation method, analyzed user's intentions by monitoring his/her activities in current navigation session and generates list of movies based on user's intentions. Mood based content navigation system has following three stages.

1. **Impulse capture:** In MediaCaddy, mood is defined as user's intentions in current session. MediaCaddy logs user's every interaction with system, between login and logout (or timeout)
2. **Intelligent knowledge extraction or mood analysis:** During this stage, user's interaction with content guide is analyzed and knowledge is extracted. During this knowledge extraction process, any of the user's clicks which were directly or indirectly related to any movie domain, are filtered. Once filtered, these clicks represent entity instances (name or artist, genre, director or movies) in Movie Domain.
3. **Content navigation:** All the entity instances filtered during knowledge extraction stage become entry points for content navigation in Movie domain ontology. Please refer to Fig. 4. In previous example Tom Cruise and Tom Hanks represents instances of artist entity, Action and Thriller are instances of genre entity, and Top Gun and Forest Gump represents Movie entity instances, these entity instances become entry point for Content navigation process.

## 9 Findings

One of the goals of our study was to give some elements in order to compare our approach in a future paper with the collaborative filtering based recommendation engine, especially since the main goal was to develop a content navigation system. We believed that bringing more information to the recommendation engine about all the entities would provide more relevant recommendation results. The actual system is based on a more semantically rich analysis than just the objects themselves and their consumption statistics. Those multiple ontological relationships between entities allowed more complex inferring bringing therefore more details about the actual user profile, and then the recommendation made to him, than what the user behavior tracking component only could capture. The relevance of the recommendation allowed to put them at the core of the navigation system instead of using a simple catalog browsing.

The system is designed to leverage the openness of services, thanks to a standards based way to describe them, in addition to techniques to map different contextual information such as content information or user profiles.

Nevertheless, a lack of XML usage within the entertainment industry has been observed, requiring multiple ways to aggregates the information within the ontology, making the architecture and mostly the maintenance of the system more complex. Promote the usage of semantic technologies and XML representation in general would definitely ease the creation of such a system.

Secondly, the lack of a common identification of content and related entities such as actors, directors, with one or a few generic IDs, makes the mapping of the information tricky, sometimes with a high degree of uncertainty. It basically creates incorrect recommendations. To minimize this effect, more complexity in the ontology creation and traversing had to be used.

Finally, the implementation of our system didn't take into consideration at every step the need of efficiency and speed. Most of the computation is done every time, after a very fast information retrieval from the ontology. We found that caching or storing the intermediate computation is possible and would bring more responsiveness. For example, intermediate results could be put back into the ontology to create more weighted relationships between entities that would be used in the traversing, which is in this case optimized. In our case, it is particularly true concerning the user related information and relationships.

## 10 Next Steps

A first version of this prototype is completed. For next steps, the entertainment industry is vast and consists of many facets that are not represented in this solution yet. A next step will be to incorporate additional entity classes to expand the size of the ontology. (e.g. different kind of content) and more relationships (e.g. more granular information about the content).

The current system doesn't use OWL [5]. Enhancing, importing and exporting the ontology using the OWL format is a one of the necessary next step to make the system more open and easier to maintain in the long run.

In this goal of making the system more open, use of standards to represent the user profile information (in a general meaning) will be studied. Schematic representations such as FOAF, OPML or attention.xml [6] are tracks that will be followed to understand how to import and also export user related information.

From a long term perspective, context extraction of a particular piece of information helps indexing this information. Examples of such techniques exist to auto categorize web pages for instance, based on the content of the page itself. Similar approach would be explored for the entities categorization within the ontology.

Additionally, the definition of categories needs more analysis. A bottom up community approach will definitely be considered. This approach would require more analysis and processing of the information. The goal would be to develop automatic, and dynamic if possible, system for a particular context. Folksonomy in a specific community is an interested manual process to do such a thing as a first step, especially for its contextual and distributed effort aspects that reduces the resources needed and creates quickly large corpus of valuable information.

## References

1. Netflix: <http://www.netflix.com>
2. Tivo: <http://www.tivo.com>
3. A. Sheth and C. Ramakrishnan, "Semantic (Web) Technology in Action": Ontology Driven Information Systems for Search, Integration and Analysis, In IEEE Data Engineering Bulletin, Special issue on Making the Semantic Web Real, December 2003.
4. International Multimedia Conference: Multimedia information services enabling: an architectural approach: Proceedings of 2001 ACM workshops on Multimedia Pages 18-23
5. Web Ontology Language (OWL):: <http://www.w3.org/2004/OWL/>
6. attention.xml: <http://developers.technorati.com/wiki/attentionxml>